



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Precontrol óptimo. Caracterización, análisis y técnicas de resolución

Vicent Giner Bosch
TESIS DOCTORAL

Directores:
Dra. D.^a Susana San Matías Izquierdo
Dr. D. Andrés Carrión García
Departamento de Estadística e Investigación
Operativa Aplicadas y Calidad

Diciembre 2014



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Precontrol óptimo. Caracterización, análisis y técnicas de resolución

Tesis doctoral presentada por
Vicent Giner Bosch

Dirigida por los doctores
D.^a Susana San Matías Izquierdo
D. Andrés Carrión García

Departamento de Estadística e Investigación
Operativa Aplicadas y Calidad

Diciembre 2014

Precontrol óptimo. Caracterización, análisis y técnicas de resolución
Tesis doctoral / Tesi doctoral / Ph.D. Thesis

Autor: Vicent Giner Bosch
Directores: Susana San Matías Izquierdo
Andrés Carrión García
Universidad: Universitat Politècnica de València
Departamento: Estadística e Investigación Operativa
Aplicadas y Calidad
Programa de doctorado: Estadística e Investigación Operativa
Aplicadas y Calidad
ID de tesis UPV: 7008
Aprobación del proyecto: 30 de junio de 2010
Depósito: 18 de noviembre de 2014
Defensa: Valencia, 22 de diciembre de 2014

Cítese como / Citeu com / Cite as:

Giner-Bosch V (2014). *Precontrol óptimo. Caracterización, análisis y técnicas de resolución*. Tesis doctoral. Departamento de Estadística e Investigación Operativa Aplicadas y Calidad, Universitat Politècnica de València, València.

Versión electrónica completa disponible en / Versió electrònica completa disponible en / Full electronic version available at:

<http://hdl.handle.net/10251/46373>

Resumen

El precontrol es una herramienta para el control de la calidad que tiene como objeto evaluar la aptitud de un proceso para producir piezas dentro de unas especificaciones dadas. Es usada, entre otros contextos, en la validación de la puesta a punto de procesos con ciclos de ejecución cortos. Basa su funcionamiento en la comparación de las sucesivas observaciones individuales muestreadas frente a los límites de tolerancia y los llamados límites de precontrol, y utiliza conteos acumulados para emitir un juicio sobre la validez del proceso. Es una técnica sencilla, rápida y fácil de aplicar.

El hecho de que, en su versión original, la técnica del precontrol no tenga en cuenta la variabilidad natural del proceso objeto de monitorización, unido a la sencillez de su diseño, provocan que, bajo ciertas circunstancias, su comportamiento no sea tan eficaz como sería deseable, en términos de la tasa de falsa alarma y de la potencia para detectar desviaciones inaceptables, hasta el punto de que algunos autores rechazan su uso. En este sentido, existen en la literatura diversas propuestas encaminadas a mejorar el comportamiento de la técnica original mediante la modificación, en mayor o menor medida, de sus reglas de funcionamiento.

En esta tesis doctoral se aborda la mejora del precontrol como herramienta para la cualificación de procesos, a través de la determinación óptima de los parámetros que rigen su comportamiento, utilizando técnicas de programación no lineal entera mixta. El objetivo es desarrollar una metodología para automatizar el proceso de elección del valor de los

parámetros del precontrol, de manera que éste se ajuste lo mejor posible a los requerimientos del usuario. Este enfoque de Optimización, inédito en el precontrol, ha sido aplicado con éxito a otras técnicas de control de la calidad en las últimas décadas, permitiendo obtener mejores herramientas para la toma de decisiones.

Para ello, en primer lugar se realiza una revisión exhaustiva de las diferentes propuestas existentes en la literatura en torno al precontrol.

Posteriormente, se define el problema a resolver y se desarrolla un modelo de programación matemática específico para la obtención de los denominados *planes de precontrol óptimos*. Se realiza una completa experiencia numérica con *software* de optimización de propósito general, mediante la que se evidencia la eficacia del modelo propuesto y, al mismo tiempo, se pone de manifiesto la existencia de diferencias significativas entre los resultados proporcionados por los diferentes algoritmos tomados en consideración, debidas en parte a la doble naturaleza no lineal y entera del problema, así como la incapacidad de dichos algoritmos para garantizar la convergencia a un óptimo global, a causa de la existencia de no convexidades.

Todo ello justifica el diseño de algoritmos específicos para la obtención de planes de precontrol óptimos, tarea que también se aborda en la presente tesis doctoral. Previamente, se realiza un estudio de las propiedades matemáticas del modelo de optimización construido, que permite ampliar el conocimiento sobre el *problema del precontrol óptimo*.

Este conocimiento es empleado en la elaboración de un algoritmo *exacto* para la resolución de dicho problema, denominado OPCenum, el cual combina una estrategia enumerativa implícita con una búsqueda local basada en métodos de búsqueda de raíces sin uso de derivadas. El algoritmo es implementado y probado sobre la misma batería de ejemplos que conformaban la experiencia numérica anterior.

Las pruebas revelan la eficacia y eficiencia del algoritmo OPCenum como método para la obtención de planes de precontrol óptimos a partir de unos requerimientos dados.

La elaboración de una interfaz gráfica para hacer más accesible el

algoritmo al usuario final, la adaptación del problema a funciones de distribución asimétricas y a ámbitos como el de la fiabilidad, así como el desarrollo de un enfoque paralelo al acometido aquí en el que se considere el precontrol como herramienta para validar la capacidad del proceso destacan como líneas futuras de investigación que surgen a partir de los resultados obtenidos en la presente tesis doctoral.

Palabras clave: *precontrol, control de calidad, cualificación de procesos, optimización, programación no lineal entera mixta, búsqueda enumerativa, métodos de búsqueda de raíces.*

Resum

El precontrol és una eina per al control de la qualitat que té com a objecte avaluar l'aptitud d'un procés per a produir peces dins d'unes especificacions donades. S'usa, entre d'altres contextos, en la validació de la posada al punt de processos amb cicles d'execució curts. El funcionament d'aquesta eina es basa en la comparació de les successives observacions individuals mostrejades davant els límits de tolerància i els anomenats límits de precontrol, i utilitza compteigs acumulats per emetre un juí sobre la validesa del procés. És una tècnica senzilla, ràpida i fàcil d'aplicar.

El fet que, en la versió original, la tècnica del precontrol no tinga en compte la variabilitat natural del procés objecte de monitoratge, unit a la senzillesa del disseny d'aquesta tècnica, provoquen que, sota certes circumstàncies, el seu comportament no siga tan eficaç com seria desitjable, en termes de la taxa de falsa alarma i de la potència per a detectar desviacions inacceptables, fins al punt que alguns autors en rebutgen l'ús. En aquest sentit, hi ha en la literatura diverses propostes encaminades a millorar el comportament de la tècnica original mitjançant la modificació, en major o menor mesura, de les seues regles de funcionament.

En aquesta tesi doctoral s'aborda la millora del precontrol com a eina per a la qualificació de processos, a través de la determinació òptima dels paràmetres que en regeixen el comportament, utilitzant tècniques de programació no lineal entera mixta. L'objectiu és desenvolupar una metodologia per a automatitzar el procés de tria del valor dels paràmetres

del precontrol, de manera que aquest s'ajuste tan bé com siga possible als requeriments de l'usuari. Este enfocament d'optimització, inèdit en el precontrol, s'ha aplicat amb èxit a altres tècniques de control de la qualitat en les últimes dècades, i ha permés obtenir millors eines per a la presa de decisions.

Amb aquest objectiu, en primer lloc es fa una revisió exhaustiva de les diferents propostes que hi ha en la literatura al voltant del precontrol.

Posteriorment, es defineix el problema que es pretén resoldre i es desenvolupa un model de programació matemàtica específic per a l'obtenció dels anomenats *plans de precontrol òptims*. S'hi porta a terme una completa experiència numèrica amb programari d'optimització de propòsit general, mitjançant la qual s'evidencia l'eficàcia del model proposat i, alhora, es posa de manifest l'existència de diferències significatives entre els resultats proporcionats pels diferents algoritmes que s'han considerat, diferències degudes en part a la doble naturalesa no lineal i entera del problema, així com la incapacitat d'aquests algoritmes per a garantir la convergència a un òptim global, a causa de l'existència de no-convexitats.

Tot això justifica el disseny d'algoritmes específics per a l'obtenció de plans de precontrol òptims, que és una de les tasques que també s'aborda en la present tesi doctoral. Prèviament, es fa un estudi de les propietats matemàtiques del model d'optimització construït, el qual permet ampliar el coneixement sobre el *problema del precontrol òptim*.

Aquest coneixement s'empra en l'elaboració d'un algoritme *exacte* per a la resolució del problema, que anomenem OPCenum i que combina una estratègia enumerativa implícita amb una cerca local basada en mètodes de cerca d'arrels sense ús de derivades. L'algoritme és implementat i provat sobre la mateixa bateria d'exemples que formaven part de l'experiència numèrica anterior.

Les proves revelen l'eficàcia i eficiència de l'algoritme OPCenum com a mètode per a obtenir plans de precontrol òptims a partir d'uns requeriments donats.

L'elaboració d'una interfície gràfica per a fer més accessible l'algoritme a l'usuari final, l'adaptació del problema a funcions de distribució

asimètriques i a àmbits com el de la fiabilitat, així com el desenvolupament d'un enfocament paral·lel al que s'ha adoptat ací, en el qual es considere el precontrol com una eina per a validar la capacitat del procés, destaquen com a línies futures de recerca sorgides a partir dels resultats obtinguts en la present tesi doctoral.

Paraules clau: *precontrol, control de qualitat, qualificació de processos, optimització, programació no lineal entera mixta, cerca enumerativa, mètodes de cerca d'arrels.*

Abstract

Pre-control is a quality control tool which assesses the ability of a process to produce pieces within given specifications. One of the contexts in which it is used is the set-up approval of short-run processes. It works by checking successive sampled observations against the tolerance limits and the so-called *pre-control limits*, using cumulative counts to yield a conclusion about the suitability of the process. It is a simple, quick technique which is easy to apply.

The fact that in its original form, the technique of pre-control does not take the natural variability of the process being monitored into account, together with its simple design, mean that its behaviour is not as effective as is desirable under certain circumstances, in terms of the false alarm rate and the power to detect unacceptable deviations, which leads some authors to reject its use. For this reason, there are different proposals in the literature focused on improving the behaviour of the original technique by modifying its operating rules to a greater or lesser extent.

The enhancement of pre-control as a tool for process qualification is addressed in this Ph.D. thesis through the optimal determination of the parameters governing its behaviour, using Mixed-Integer Nonlinear Programming techniques. The goal is to develop a methodology to automatize the procedure of selecting the value of the pre-control parameters in order to satisfy the user's requirements as closely as possible. This Optimisation approach, unprecedented in pre-control, has been successfully applied to other quality control techniques in recent

decades, thereby producing better decision-making tools.

To achieve this goal, firstly an exhaustive review of the different proposals regarding pre-control in the literature is outlined.

The problem to be solved is then defined and a Mathematical Programming model which is specifically designed to obtain the so-called *optimal pre-control plans* is developed. A complete numerical experiment using general-purpose optimisation software is described, showing the effectiveness of the proposed model and, at the same time, revealing the existence of significant differences between the results provided by the different algorithms being considered, due in part to the two-fold nature—nonlinear and integer—of the problem, and also the inability of these algorithms to guarantee convergence to a global optimum, due to the existence of nonconvexities.

All of these things justify the design of specific algorithms to obtain optimal pre-control plans, which is another goal of this Ph.D. thesis. Before this, a study of the mathematical properties of the optimization model which was previously built is outlined, improving knowledge about the *optimal pre-control problem*.

This knowledge is used in the development of an *exact* algorithm called OPCenum to solve the problem, combining an implicit enumerative strategy with a local search based on derivative-free root-finding methods. The algorithm is implemented and tested on the same instances which were considered in the previous numerical experiment.

The tests reveal the effectiveness and efficiency of the algorithm OPCenum as a method to obtain optimal pre-control plans for some given requirements.

The development of a graphical interface to make the algorithm more user-friendly, the adaptation of the problem to asymmetric distribution functions and to other fields such as reliability, and the development of a complementary approach where pre-control is considered as a tool to assess process capability are the main future research lines arising from the results obtained in this Ph.D. thesis.

Keywords: *Pre-control, quality control, process qualification, optimisation, mixed-integer nonlinear programming, enumerative search, root-finding methods.*

La investigación es consubstancial con la universidad. No se entiende una universidad sin generación de conocimiento. ¿Qué seríamos sin cumplir esa faceta? ¿Una academia de apuntes superiores?

—Justo Nieto, Catedrático de Cinemática y Dinámica de Máquinas,
Rector de la Universitat Politècnica de València desde 1986
hasta 2004.
Polivalencia, 28 (2004).

Una universitat que no fa investigació científica o tecnològica, no és universitat.

—Vicente Conejero, Catedrático de Bioquímica y Biología Molecular, Director hasta 2008 del Instituto de Biología Molecular y Celular de Plantas, centro de investigación mixto de la Universitat Politècnica de València y el Consejo Superior de Investigaciones Científicas.
Bertrana, U. et al. *Un passeig pel campus. Reflexions sobre la societat i el coneixement de personalitats científiques universitàries*. Publicacions i edicions de la Universitat de Barcelona, Barcelona (2008)

Sincerament, em considere afortunat de tindre aquesta professió i crec que sóc deutor de la societat, que amb els seus impostos fa que hi haja universitats i professors investigadors en ciència [...] que se'ns paga per pensar, per cavil·lar i per aportar la nostra contribució a un avançament del coneixement que després tindrà efectes positius per a la nostra vida quotidiana.

—Ramon Lapiedra, Catedrático de Física Teórica, Rector de la Universitat de València desde 1984 hasta 1994.
Ibíd.

Agraïments

*Thank you for holding me, [...]
thank you for helping me.*

—Sinéad O'Connor.

*Thank You for Hearing Me,
Universal Mother (1994)*

Quan pense en tot el treball que hi ha darrere d'esta tesi doctoral, que ara —per fi— presente, em ve al cap, directament des dels meus records de la infància, *la portada clàssica de l'àlbum «Astèrix i Cleopatra» de R. Goscinny i A. Uderzo (1965)*, en la qual, per caricaturitzar les grans superproduccions cinematogràfiques de tall èpic de l'època, s'inclou el text següent:

*La més gran aventura mai dibuixada. 14 litres de tinta xinesa,
30 pinzells, 62 llapis de mina tova, 1 llapis de mina dura, 27 gomes
d'esborrar, 38 quilos de paper, 16 cintes de màquina d'escriure,
2 màquines d'escriure i 62 litres de cervesa foren necessaris per a la
seua realització!*

El fet és que, igual que aquelles *superproduccions*, esta tesi no hauria sigut possible sense l'ajuda i la col·laboració de moltíssima gent. Ho diré més clarament agafant prestades les paraules del meu col·lega de professió Thijs Urlings, amb qui vaig coincidir en la Universitat Politècnica de València, i qui en els agraïments de *la seua tesi doctoral (2010)* diu açò:

A pesar que podria paréixer diferent a aquells qui no s'hi dediquen, treballar en ciència no és una qüestió individual. Deixant a banda el fet que u sempre ha de tindre en compte la literatura relacionada i el treball d'altres, ningú aconsegueix acabar una tesi doctoral sense l'ajuda i el suport del seu entorn professional i personal.

Per això, vull començar amb unes paraules de record per a Susana San Matías, directora d'esta tesi, però també companya, consellera i amiga, qui va faltar tràgicament en agost de 2012, quan ja teníem enllestida una bona part del treball. Susana era excepcional com a persona i com a professional, i em sent afortunat per haver pogut conèixer-la i treballar amb ella. Este document porta la seua empremta en cada pàgina. No podré mai agrair-li suficientment la seua implicació, la qual va anar sempre moltíssim més enllà de l'estrictament necessari, la seua cordial complicitat i tot el temps, l'esforç i la paciència que va dedicar a esta tesi i a la meua formació com a investigador. Sé que la seua manera de ser i fer en el treball ha calat en mi, i n'estic orgullós.

També vull agrair d'una manera molt especial a Andrés Carrión el seu esforç com a tutor i codirector de la tesi. A banda dels seus inestimables coneixements en control de la qualitat, ell ha aportat sentit comú, tranquil·litat i ànims en la recta final de l'elaboració d'este treball i dels tràmits relacionats amb la seua presentació.

Durant tot este temps de doctorat, he recordat moltes voltes els meus professors de la Facultat de Matemàtiques de la Universitat de València. D'alguna forma, tinc la sensació que estic culminant un procés que va començar allí l'any 1992. A ells vull agrair-los la bona formació que hi vàrem rebre; en particular, la capacitat per a estructurar, formalitzar i abordar de manera metòdica, rigorosa i amb mentalitat oberta la resolució de tota classe de problemes. Puc dir que han sigut eines molt útils en el meu treball quotidià i en esta tesi en concret. Així mateix, vull aprofitar per a agrair-los el tracte professional, pròxim, senzill i humà que ens varen donar, i que vull pensar que he sabut traslladar a la meua activitat com a docent.

Agraïsc de manera molt humil i sincera el recolzament que he rebut durant estos anys de doctorat per part dels meus companys del Departament d'Estadística i Investigació Operativa Aplicades i Qualitat de la Universitat Politècnica de València. És el moment de reconèixer que en ocasions he abusat de la seua generositat i professionalitat. Vull agrair en especial a Rubén Ruiz els seus consells de tota classe, sempre ràpids i encertats, i en especial els referits a l'avaluació d'algoritmes realitzada en les experiències numèriques dels capítols 3 i 5 d'esta tesi; a Sebastià Balasch i Rosa Alcover, els precisos aclariments sobre la tècnica de l'anàlisi de la variància en relació a les mencionades experiències; a Juan Carlos García, eixes converses filosòfiques interminables i també els consells pràctics sobre l'ARL i altres històries; a Ángeles Calduch, eixes hores de treball que m'ha estalviat; a Eva Vallada, la seua companyia silenciosa i el seu consell sempre sincer; a Mónica Clemente com a fidel *companya d'aventures*; a Ana I. Sánchez, la calma i proximitat que transmet com a directora del departament; a Ana Aparicio per fer fàcils tots els tràmits burocràtics de la tesi; i, en general, tots els ànims que he rebut de manera contínua per part de molta gent, i que m'han ajudat a avançar.

Vull donar gràcies també al meu excompany d'estudis F. Javier Toledo, de la Universitat Miguel Hernández, per la inspiració que em va proporcionar en un moment de bloqueig en relació a una de les demostracions del capítol 4 d'esta memòria.

En l'aspecte tècnic, he de reconèixer l'ajuda que he obtingut de diverses persones a través de fòrums d'Internet especialitzats en GAMS, C++, GSL, L^AT_EX, matemàtiques, estadística, etc. Agraïsc als col·legues ja mencionats Rubén Ruiz i Thijs Urlings que em facilitaren les seues plantilles de L^AT_EX per a tesis, les quals varen resultar-me realment útils. Gràcies al meu amic **Ramón Roselló** pel seu excel·lent treball en el disseny de la coberta de la tesi. També quede molt agraït al Servei de Promoció i Normalització Lingüística de la Universitat Politècnica de València i a **Craig D'Souza** per haver revisat la traducció al valencià i a l'anglès, respectivament, del resum d'esta tesi.

Per últim, agraiïsc als professors Enriqueta Vercher, Ramón Álvarez-Valdés i Maite León, de la Universitat de València, Salvador Naya, de la Universidade da Coruña, i Rubén Ruiz i Juan Carlos García, de la Universitat Politècnica de València, els seus encertats comentaris i suggerències, bé com a avaluadors externs o bé com a membres — titulars o suplents— del tribunal de la present tesi doctoral, els quals han contribuït, sense dubte, a millorar-la de manera significativa.

En l'àmbit personal, per damunt de tot vull donar les gràcies sincerament als meus pares per haver sigut un pilar segur i ferm per a mi. Sense ells, hui no estaria escrivint estes línies. Ells m'han transmés, a través del seu exemple, l'amor pel treball i per la ciència. Em varen donar llibertat per a triar els meus estudis i el meu camí en la vida, i m'han demostrat contínuament la seua confiança. De mon pare, Vicente, he après que es pot buscar l'excel·lència en el treball sense deixar de ser senzill i pròxim. De ma mare, Lolita, he après que la dedicació i el sacrifici poden tindre sentit i donar fruit. Dels dos he après que ser honrat i tractar bé a la gent és lo primer i que sempre acaba eixint a compte. Sense dubte, han influït en la meua vida i en el meu treball més de lo que a voltes es pensen.

Faig extensiu este agraïment a tota la meua família, en especial a la meua germana, Lola, a la qual em sent molt unit. No s'ha cansat mai d'animar-me i recolzar-me, tot aportant-hi el contrapunt realista o somniador que necessitava en cada moment. Sempre ha cregut en mi, i jo en ella. Això és lo que ens fa germans de veres.

A Jana, la meua companya de vida, he d'agrair-li que haja sabut esperar, recolzar, patir... moltes vegades per damunt del seu *límit de control superior*; que s'haja sacrificat més d'allò que jo puga arribar a imaginar. I també vull agrair-li que haja compartit amb mi la seua visió de la vida des de l'enginyeria, encaminada sempre a buscar solucions pràctiques a qualsevol problema que es pose per davant.

Agraiïsc als meus amics el seu suport incondicional i il·limitat. Realment, en alguns moments han cregut que podia acabar la tesi més que jo mateix; i quan no ho han pensat, no m'ho han dit. Gràcies a Maria per fer-me comprendre que ser professor sense tindre la tesi és hui en dia un

privilegi. Gràcies a Olga per ser una gran *coach*. Gràcies a Loli i Adriana per mantindre l'interés des de la distància.

Finalment, agraïsc a tots, família, amics i companys, la paciència que m'heu demostrat en tots estos anys que he estat enganxat al doctorat (oficialment, des de 2003). Lamente haver-vos furtat massa temps d'estar junts, i espere que sabreu disculpar-ho. Com diu **una cançó dels Manel**, «ens ha costat Déu i ajuda arribar fins ací». Ara acaba tot; ara comença tot.

València, novembre de 2014

Índice general

1. Introducción y objetivos	1
1.1. Control y mejora de la calidad. Definiciones básicas	2
1.1.1. Calidad, producto, características de calidad y proceso	2
1.1.2. Satisfacción versus conformidad	3
1.1.3. Planificación, control y mejora de la calidad	4
1.1.4. Control estadístico de la calidad	5
1.2. Control estadístico de procesos	6
1.2.1. Variabilidad, estabilidad y control	6
1.2.2. Gráficos de control	8
1.2.3. Análisis de la capacidad de un proceso	19
1.3. Precontrol	24
1.4. Optimización en control de la calidad	25
1.4.1. Programación no lineal entera mixta	26
1.4.2. Métodos de resolución de MINLP	28
1.5. Objetivos y estructura de la tesis	39
2. Precontrol. Origen y estado actual	43
2.1. La técnica del precontrol	43
2.1.1. Definición y origen del precontrol	43
2.1.2. Precontrol clásico	44
2.1.3. Aplicación del precontrol	49
2.1.4. Ventajas y puntos débiles del precontrol clásico	51

2.2.	Alternativas al precontrol clásico	55
2.2.1.	Precontrol en dos etapas	57
2.2.2.	Precontrol modificado	58
2.2.3.	Precontrol de las diez unidades	61
2.2.4.	Precontrol del cambio de media	61
2.2.5.	Precontrol simplificado	63
2.2.6.	<i>Pairs</i>	66
2.2.7.	Precontrol modificado de Družovec <i>et al.</i>	67
2.2.8.	<i>Set-Up Process Algorithm</i>	69
2.2.9.	Resumen de las variantes del precontrol	70
2.3.	Otras contribuciones recientes	71
2.3.1.	Monitorización de parámetros del proceso con precontrol	72
2.3.2.	Precontrol sobre el porcentaje de tolerancias	73
2.3.3.	Gráfico de sectores	74
2.3.4.	Aplicación del SUPA a la puesta a punto de proc.	75
2.3.5.	Precontrol multivariante	77
2.4.	Precontrol generalizado	80
2.5.	Evaluación del comportamiento de los planes de precontrol generalizados	82
2.5.1.	Precontrol como contraste sobre la media	82
2.5.2.	Precontrol como contraste sobre la capacidad	92
2.6.	Un ejemplo práctico	96
2.7.	Conclusiones	99
3.	Precontrol óptimo	101
3.1.	Introducción	101
3.2.	Planteamiento de la determinación de los parámetros del precontrol como problema de optimización	102
3.2.1.	Variables del modelo	103
3.2.2.	Criterios de optimización	103
3.2.3.	Restricciones a contemplar	111

3.3. Desarrollo del modelo de optimización para el precontrol generalizado	112
3.3.1. El modelo general	112
3.3.2. Hipótesis y notación	113
3.3.3. Obtención de la función objetivo	114
3.3.4. Obtención de las restricciones	119
3.4. El precontrol óptimo	121
3.5. Resolución del modelo del precontrol óptimo. Discusión y experiencia numérica	123
3.5.1. Estrategias y herramientas de resolución	124
3.5.2. Experiencia numérica	125
3.5.3. Análisis del peor caso	196
3.6. Conclusiones	199
4. Estudio de las propiedades del modelo de optimización del precontrol	201
4.1. Notación, objetivos y estructura	202
4.1.1. Notación	202
4.1.2. Objetivos, justificación y estructura	208
4.2. Algunos resultados básicos	212
4.3. Propiedades de las funciones del precontrol óptimo	215
4.3.1. Comportamiento como función de λ	215
4.3.2. Comportamiento como función de k	252
4.3.3. Comportamiento como función de t	258
4.3.4. Comportamiento asintótico como función de k y t	264
4.4. Optimización del valor de λ , fijados k y t	266
4.4.1. Optimización local de λ	268
4.4.2. Consecuencias en la resolución del problema del precontrol óptimo	279
4.5. Extensión a otras distribuciones de probabilidad	287
4.6. Conclusiones	289

5. Un algoritmo exacto para la determinación de planes de precontrol óptimos	291
5.1. Introducción	291
5.2. Estrategias de búsqueda de soluciones	293
5.2.1. Mejora <i>local</i> del parámetro λ	293
5.2.2. Enumeración implícita de las soluciones	298
5.3. Algoritmo de mejora local de λ	299
5.3.1. Ejemplos de aplicación	301
5.4. Algoritmo de enumeración implícita para el problema del precontrol óptimo	305
5.4.1. Parámetros del problema y del algoritmo	308
5.4.2. Orden de visita de las soluciones	309
5.4.3. Reducción de la zona de búsqueda	310
5.4.4. Descripción del algoritmo	311
5.4.5. Ejemplo numérico	316
5.5. Implementación de los algoritmos	320
5.5.1. Detalles de la implementación	321
5.5.2. Uso de bibliotecas externas	325
5.6. Experiencia numérica	326
5.6.1. Resultados numéricos	327
5.6.2. Comparación con los resultados de aplicar de <i>software</i> de propósito general	334
5.7. Conclusiones	335
6. Otras contribuciones y producción científica asociada	337
6.1. Otras aportaciones al precontrol	337
6.1.1. Análisis de sensibilidad ante la ausencia de normalidad	337
6.1.2. Enfoques no paramétricos	339
6.1.3. Aplicación en el ámbito de la fiabilidad	339
6.1.4. Otras estrategias de resolución	350
6.2. Contribuciones en el ámbito científico	350
6.2.1. Publicaciones en revistas científicas indexadas	351

6.2.2.	Publicaciones en congresos internacionales ind.	351
6.2.3.	Comunicaciones a congresos internacionales	352
6.2.4.	Comunicaciones a congresos nacionales	353
6.2.5.	<i>Technical reports</i>	354
6.2.6.	Trabajos académicos	354
7.	Conclusiones y líneas futuras de investigación	355
7.1.	Conclusiones	355
7.2.	Líneas futuras de investigación	358
7.2.1.	Desarrollo de una interfaz gráfica	359
7.2.2.	Estudio del peor caso	361
7.2.3.	Adecuación del modelo a otros requerimientos	362
7.2.4.	Precontrol óptimo como contraste sobre la capac.	362
7.2.5.	Análisis de sensibilidad ante la ausencia de norma- lidad	363
7.2.6.	Enfoques no paramétricos	363
7.2.7.	Adaptación a otras distribuciones de probabilidad	364
7.2.8.	Aplicación en el ámbito de la fiabilidad	364
7.2.9.	Otras estrategias de resolución	365
7.2.10.	Precontrol óptimo multiobjetivo	367
7.2.11.	Precontrol óptimo multivariante	367
7.2.12.	Extensión a otras herramientas de control estadístico de la calidad	368
	Anexos	369
	A. Resultados de la experiencia numérica con Excel y GAMS	371
	Bibliografía	399

Índice de figuras

1.1. Ejemplo de gráfico de control \bar{X} - R	10
1.2. Relación de capacidad para tres procesos con diferente dispersión	22
1.3. Diferencia entre capacidad potencial y real	23
2.1. Método de clasificación del precontrol clásico	45
2.2. Esquema de aplicación del precontrol clásico	49
2.3. Carátula para la esfera de un reloj comparador analógico	50
2.4. Ejemplo de gráficos de precontrol	51
2.5. Método de clasificación del precontrol del cambio de media	63
2.6. Método de clasificación del precontrol simplificado	65
2.7. Método de clasificación del precontrol modificado de Družovec et al.	68
2.8. Zonas verde, amarilla y roja en el precontrol multivariante	79
2.9. Probabilidades de obtener una pieza verde, amarilla y roja	89
2.10. Curvas características del precontrol como contraste sobre δ , con $t = 2$ y $\alpha = 12,29\%$, para un proceso con $C_p = 1$	90
2.11. Curvas características del precontrol como contraste sobre δ , con $k = 5$ y $\alpha = 12,29\%$, para un proceso con $C_p = 1$	90
2.12. Curvas características del precontrol como contraste sobre δ , para diferentes valores de k y t , y $\alpha = 12,29\%$, para un proceso con $C_p = 1$	91

2.13. Curvas características del precontrol como contraste sobre δ , para diferentes valores de k y t , y $\alpha = 2,5\%$, para un proceso con $C_p = 1$	91
2.14. Curvas características del precontrol como contraste sobre C_p , con $t = 2$ y $\alpha = 12,29\%$, para un proceso con $\delta' = 0$	94
2.15. Curvas características del precontrol como contraste sobre C_p , con $k = 5$ y $\alpha = 12,29\%$, para un proceso con $\delta' = 0$	94
2.16. Curvas características del precontrol como contraste sobre C_p , para diferentes valores de k y t , y $\alpha = 12,29\%$, para un proceso con $\delta' = 0$	95
2.17. Curvas características del precontrol como contraste sobre C_p , para diferentes valores de k y t , y $\alpha = 2,5\%$, para un proceso con $\delta' = 0$	95
2.18. Croquis de una biela	97
3.1. Mínimo número esperado de piezas a muestrear para cada ejemplo de la batería en la experiencia numérica de la subsección 3.5.2	145
3.2. Calibración de GAMS/AlphaECP. Efectos principales	157
3.3. Calibración de GAMS/AlphaECP. Interacciones	158
3.4. Calibración de GAMS/DICOPT. Análisis de las ejecuciones sin solución	163
3.5. Calibración de GAMS/DICOPT. Efectos principales	165
3.6. Calibración de GAMS/DICOPT. Interacciones	166
3.7. Calibración de GAMS/OQNLP. Efectos principales	168
3.8. Calibración de GAMS/OQNLP. Interacciones	169
3.9. Calibración de GAMS/SBB. Efectos principales	173
3.10. Calibración de GAMS/SBB. Interacciones	174
3.11. Comparación de los algoritmos. Efectos principales	177
3.12. Comparación de los algoritmos. Interacciones	178
3.13. Calibración de GAMS/AlphaECP. Caso $C_p = 1,2$. Efectos principales	184

3.14. Calibración de GAMS/AlphaECP. Caso $C_p = 1,2$. Interacciones	185
3.15. Calibración de GAMS/DICOPT. Caso $C_p = 1,2$. Efectos principales	187
3.16. Calibración de GAMS/DICOPT. Caso $C_p = 1,2$. Interacciones	188
3.17. Calibración de GAMS/OQNLP. Caso $C_p = 1,2$. Efectos principales	189
3.18. Calibración de GAMS/SBB. Caso $C_p = 1,2$. Efectos principales	190
3.19. Comparación de los algoritmos. Efectos principales. Caso $C_p = 1,2$	193
3.20. Curva de $E[n(\delta)]$, fijado el plan de precontrol óptimo, para $\alpha^* = 1\%$, $\beta^* = 0,5\%$ y δ^* tal que $p^* = 2\%$	198
3.21. Relación del precontrol óptimo con propuestas anteriores	199
4.1. Probabilidades de pieza verde y amarilla en función del parámetro λ	215
4.2. Comportamiento de la función $V(\lambda)$	219
4.3. Comportamiento de α como función de λ	228
4.4. Comportamiento de β como función de λ	228
4.5. Comportamiento del polinomio $P(x)$	230
4.6. Comportamiento del polinomio $Q(x)$	232
4.7. Comportamiento de la función $R^{(m)}(x)$	234
4.8. Comportamiento de las funciones $C^{(m)}(\lambda)$ y $D^{(m)}(\lambda)$	237
4.9. Comportamiento de f como función de λ	242
4.10. Caracterización del máximo de f como función de λ	249
4.11. Comportamiento de α como función de k	256
4.12. Comportamiento de β como función de k	256
4.13. Comportamiento de f como función de k	258
4.14. Comportamiento de α como función de t	262
4.15. Comportamiento de β como función de t	262
4.16. Comportamiento de f como función de t	264
4.17. Ejemplo de determinación de λ_α	270

4.18. Ejemplo de determinación de λ_β	272
4.19. Ejemplo de determinación de λ'	277
4.20. Resolución del problema OPC-sb. Ejemplo 1	282
4.21. Resolución del problema OPC-sb. Ejemplo 2	283
5.1. Ejemplo de determinación de λ_α y λ_β . Caso factible	306
5.2. Ejemplo de determinación de λ_α y λ_β . Caso no factible	307
5.3. Tasa de llamadas a la función α por cada iteración del algoritmo OPCenum en la experiencia numérica de la sección 5.6	332
5.4. Tasa de llamadas a la función β por cada iteración del algoritmo OPCenum en la la experiencia numérica de la sección 5.6	333
6.1. Definición de las zonas verde, amarilla y roja para un proceso exponencial	342
6.2. Gráfico de precontrol para la simulación de un sistema con un tiempo medio entre fallos de $\mu_0 = 1000$ horas realizada en la subsección 6.1.3	349
6.3. Gráfico de precontrol para la simulación de un sistema con un tiempo medio entre fallos de $\mu_1^* = 100$ horas realizada en la subsección 6.1.3	349

Índice de tablas

2.1. Valores de α y β_{δ^*} con $\lambda = 4$, para el ejemplo de la sec. 2.6	99
2.2. Valores de β_{δ^*} y λ con $\alpha = 0,01$, para el ejemplo de la sección 2.6	99
3.1. Relación de los algoritmos considerados en la experiencia numérica de la subsección 3.5.2	134
3.2. Algoritmos de programación lineal entera admitidos por los diferentes <i>solvers</i> para MINLP de GAMS	135
3.3. Algoritmos de programación no lineal admitidos por los diferentes <i>solvers</i> para MINLP de GAMS	136
3.4. Cálculo del número total de variantes de algoritmos de MINLP a evaluar en la experiencia numérica de la subsección 3.5.2	137
3.5. Resumen del comportamiento de las ejecuciones realizadas sobre la batería de problemas en la experiencia numérica de la subsección 3.5.2, desde el punto de vista de los problemas	142
3.6. Listado de algoritmos con los mejores resultados en la experiencia numérica de la subsección 3.5.2	147
3.7. Listado de algoritmos a comparar en la segunda etapa de la experiencia numérica de la subsección 3.5.2	176

3.8. Resumen del comportamiento de los tres grupos de algoritmos identificados en la segunda fase de la experiencia numérica de la subsección 3.5.2	180
3.9. Resumen de la calibración de algoritmos en la experiencia numérica de la subsección 3.5.2. Caso $C_p = 1,2$	191
3.10. Estudio del peor caso para una selección de los planes de precontrol generados con $C_p = 1,3333$ y $\delta^* = 1,946$ (δ^* tal que $p^* = 2\%$)	197
4.1. Expresiones utilizadas en el capítulo 4	207
5.1. Detalle de las iteraciones correspondientes al ejemplo de la subsección 5.4.5	318
5.2. Resultados de la experiencia numérica con OPCenum	329
5.3. Tasa de llamadas a la función α por cada iteración del algoritmo OPCenum en la experiencia numérica de la sección 5.6	331
5.4. Tasa de llamadas a la función β por cada iteración del algoritmo OPCenum en la experiencia numérica de la sección 5.6	331
6.1. Primeros 45 valores simulados para $\mu = \mu_0 = 1000$ horas en la experiencia numérica de la subsección 6.1.3	348
6.2. Primeros 45 valores simulados para $\mu = \mu_1^* = 100$ horas en la experiencia numérica de la subsección 6.1.3	348
A.1. Batería de ejemplos correspondiente a la experiencia numérica de la subsección 3.5.2	373
A.2. Algoritmos probados en la experiencia numérica de la subsección 3.5.2	380
A.3. ANOVA correspondiente a la calibración del algoritmo GAMS/AlphaECP	381

A.4. Tabla de medias para cada nivel de los métodos MILP y NLP correspondiente al algoritmo GAMS/AlphaECP. Caso general	381
A.5. Tabla de medias para cada nivel de los parámetros C_p , α^* y β^* correspondiente al algoritmo GAMS/AlphaECP. Caso general	382
A.6. Tabla de medias para cada combinación de los métodos MILP y NLP correspondiente al algoritmo GAMS/AlphaECP. Caso general	383
A.7. ANOVA correspondiente a la calibración del algoritmo GAMS/DICOPT	384
A.8. Tabla de medias para cada nivel de los métodos MILP y NLP correspondiente al algoritmo GAMS/DICOPT. Caso general	384
A.9. Tabla de medias para cada unidad experimental correspondiente al algoritmo GAMS/DICOPT. Caso general	385
A.10. Tabla de medias para cada combinación de los métodos MILP y NLP correspondiente al algoritmo GAMS/DICOPT. Caso general	386
A.11. ANOVA correspondiente a la calibración del algoritmo GAMS/OQNLP	387
A.12. Tabla de medias para cada nivel del método NLP correspondiente al algoritmo GAMS/OQNLP. Caso general	387
A.13. Tabla de medias para cada nivel de los parámetros C_p , α^* y β^* correspondiente al algoritmo GAMS/OQNLP. Caso general	388
A.14. ANOVA correspondiente a la calibración del algoritmo GAMS/SBB	388
A.15. Tabla de medias para cada nivel del método NLP correspondiente al algoritmo GAMS/SBB. Caso general	389
A.16. Tabla de medias para cada nivel de los parámetros C_p , α^* y β^* correspondiente al algoritmo GAMS/SBB. Caso general	389
A.17. ANOVA correspondiente a la comparación de los algoritmos	390

A.18.ANOVA correspondiente a la calibración del algoritmo GAMS/AlphaECP. Caso $C_p = 1,2$	391
A.19.Tabla de medias para cada nivel de los métodos MILP y NLP y cada unidad experimental correspondiente al algoritmo GAMS/AlphaECP. Caso $C_p = 1,2$	391
A.20.Tabla de medias para cada combinación de los métodos MILP y NLP correspondiente al algoritmo GAMS/AlphaECP. Caso $C_p = 1,2$	392
A.21.ANOVA correspondiente a la calibración del algoritmo GAMS/DICOPT. Caso $C_p = 1,2$	393
A.22.Tabla de medias para cada nivel de los métodos MILP y NLP y cada unidad experimental correspondiente al algoritmo GAMS/DICOPT. Caso $C_p = 1,2$	393
A.23.Tabla de medias para cada combinación de los métodos MILP y NLP correspondiente al algoritmo GAMS/DICOPT. Caso $C_p = 1,2$	394
A.24.ANOVA correspondiente a la calibración del algoritmo GAMS/OQNLP. Caso $C_p = 1,2$	395
A.25.Tabla de medias para cada nivel del método NLP y cada unidad experimental correspondiente al algoritmo GAMS/OQNLP. Caso $C_p = 1,2$	395
A.26.ANOVA correspondiente a la calibración del algoritmo GAMS/SBB. Caso $C_p = 1,2$	395
A.27.Tabla de medias para cada nivel del método NLP y cada unidad experimental correspondiente al algoritmo GAMS/SBB. Caso $C_p = 1,2$	396
A.28.ANOVA correspondiente a la comparación de los algoritmos. Caso $C_p = 1,2$	396
A.29.Tabla de medias para cada algoritmo y cada unidad experimental. Caso $C_p = 1,2$	397

Índice de resultados demostrados

2.1. Proposición (Buena definición de $\Pr(\text{cualificar} \mid \delta)$)	85
4.1. Proposición (Propiedades de p_g como función de λ)	212
4.1. Corolario (Propiedades de p_y como función de λ)	213
4.1. Lema	216
4.1. Teorema (Comportamiento de la probabilidad de cualificar como función de λ)	220
4.2. Corolario (Comportamiento de α como función de λ)	226
4.3. Corolario (Comportamiento de β como función de λ)	227
4.2. Lema	229
4.3. Lema	230
4.4. Lema	232
4.5. Lema	234
4.2. Teorema (Comportamiento de la función objetivo como función de λ)	236
4.4. Corolario (Caracterización del máximo de la función objeti- vo como función de λ)	243
4.5. Corolario	248
4.6. Corolario (No convexidad del problema OPC)	250
4.2. Proposición (Comportamiento de la probabilidad de cuali- ficar como función de k)	252
4.7. Corolario (Comportamiento de α como función de k)	255

4.8. Corolario (Comportamiento de β como función de k) . . .	255
4.3. Proposición (Comportamiento de la función objetivo como función de k)	257
4.4. Proposición (Comportamiento de la probabilidad de cuali- ficar como función de t)	259
4.9. Corolario (Comportamiento de α como función de t) . . .	260
4.10. Corolario (Comportamiento de β como función de t) . . .	261
4.5. Proposición (Comportamiento de la función objetivo como función de t)	263
4.6. Proposición (Comportamiento asintótico de la probabilidad de cualificar como función de k y t)	264
4.11. Corolario (Comportamiento de α como función de k y t) .	265
4.12. Corolario (Comportamiento de β como función de k y t) .	265
4.7. Proposición (Comportamiento asintótico de la función objetivo como función de k y t)	266
4.8. Proposición (Definición de λ_α)	268
4.9. Proposición (Definición de λ_β)	271
4.3. Teorema (Optimización local de λ)	273
4.13. Corolario (Optimización del precontrol con cotas superiores para k y t)	279
4.10. Proposición	284
4.14. Corolario	285
4.15. Corolario	286

Índice de algoritmos

2.1. Precontrol clásico – Fase de cualificación	47
2.2. Precontrol clásico – Fase de Mantenimiento	48
2.3. Precontrol en dos etapas – Fase de mantenimiento	59
2.4. Precontrol de las diez unidades – Fase de mantenimiento	62
2.5. Precontrol del cambio de media – Fase de mantenimiento	64
2.6. Precontrol simplificado – Fase de mantenimiento	66
2.7. Precontrol modificado de <i>Družovec et al.</i> – Fase de cualifi- cación	68
2.8. Precontrol generalizado – Fase de cualificación	83
5.1. Función <code>OPC_improve_lambda</code>	300
5.2. Procedimiento <code>OPCenum_improve</code>	312
5.3. Procedimiento <code>OPCenum_update_status</code>	313
5.4. Procedimiento <code>OPCenum_set_next_step</code>	314
5.5. Algoritmo <code>OPCenum</code>	315

Introducción y objetivos

En el ámbito industrial y empresarial el uso de herramientas numéricas para el control, la gestión y la mejora de la producción es, hoy en día, un requisito ineludible. En particular, la búsqueda de la *calidad* de los bienes y servicios producidos se ha convertido en un factor clave a la hora de atraer y fidelizar a los clientes y, por tanto, hacer frente a los competidores.

La calidad tiene que ser medida para que sea controlable, evaluable y mejorable. Las herramientas de tipo cuantitativo empleadas a tal efecto, basadas en la Estadística y en otras disciplinas, son, al mismo tiempo, revisadas y mejoradas cada vez más habitualmente utilizando, entre otras, técnicas de Investigación Operativa.

En este capítulo presentamos, en primer lugar, los conceptos básicos que será necesario conocer en relación al control de la calidad y, en particular, al control estadístico de procesos. Posteriormente se introducirá el precontrol, una herramienta para el control de la calidad que es objeto principal de estudio y mejora en la presente tesis doctoral. Asimismo, se revisa la aplicación de herramientas de Investigación Operativa en el ámbito del control de la calidad, dedicando especial atención a las técnicas de programación no lineal entera mixta.

Todo ello permitirá contextualizar convenientemente el trabajo de investigación desarrollado, cuya estructura y objetivos serán detallados

como conclusión de este capítulo.

1.1. Control y mejora de la calidad. Definiciones básicas

Comenzamos este capítulo presentando algunos conceptos básicos de ingeniería de la calidad, con el fin de establecer el entorno en el que se sitúa el trabajo que hemos desarrollado y de introducir la nomenclatura y la notación que posteriormente utilizaremos¹.

1.1.1. Calidad, producto, características de calidad y proceso

Se entiende por *producto* el resultado o salida de un proceso, bien sea una mercancía, bien un *software* o una información, o bien un servicio.

La *calidad* puede ser definida como el conjunto de características de un producto o servicio que, con su aptitud, conducen a satisfacer una o varias necesidades, expresas o implícitas, de los clientes. También puede entenderse la calidad como la ausencia de deficiencias. [Montgomery \(2012\)](#) añade la consideración de que la calidad es inversamente proporcional a la variabilidad (dado que, como veremos después, en cierto sentido las deficiencias están ligadas a la inestabilidad o variabilidad).

Las características de un producto con las que se espera satisfacer a los clientes pueden ser de naturaleza técnica (medidas, dimensiones, etc.) o de otra índole (rapidez, facilidad de uso, adaptabilidad, atractivo, etc.).

Reciben el nombre de *características de calidad* aquellas propiedades elementales de que se componen las diferentes cualidades que a su vez conforman los componentes del producto. La mayoría de las características de calidad pueden expresarse numéricamente.

De acuerdo con la norma ISO 9000:2005 ([AENOR, 2005](#)), un *proceso* puede definirse como el «conjunto de actividades mutuamente relaciona-

¹La mayor parte de las definiciones y conceptos expuestos en esta sección y la siguiente han sido extraídos de [Juran y Gryna \(1993\)](#), [Montgomery \(2012\)](#) y [Romero Villafranca y Zúñica Ramajo \(2008\)](#).

das o que interactúan, las cuales transforman elementos de entrada en resultados», haciendo uso de determinados *recursos* (Camisión et al., 2006). En muchos casos, los procesos se encadenan, de modo que los resultados de uno son los elementos de entrada de otro u otros. Evans y Lindsay (2008) hacen énfasis sobre todo en la *creación de valor* para el cliente que va ligada a muchos procesos, si bien no a todos: es el caso de los procesos de carácter técnico, entre otros, que no crean valor directamente pero permiten que los que sí lo hacen funcionen correctamente.

Asimismo, en el contexto de control de la calidad es habitual, por abuso del lenguaje, decir que cierto proceso está *centrado* o *descentrado* (conceptos que se introducirán más adelante), o que presenta cierta distribución de probabilidad, etc., cuando realmente se quiere aludir al comportamiento o a propiedades de una de las características de calidad implicadas en dicho proceso.

1.1.2. Satisfacción versus conformidad

La *satisfacción* tiene que ver con las características del producto, y se alcanza cuando se consigue que éstas respondan a las necesidades del cliente; además, previsiblemente, hacen que éste decida adquirirlo (bien sea por primera vez o no).

La *conformidad* con los posibles procedimientos, normas o *especificaciones* existentes para el producto no debe ser tomada por la empresa como única definición de calidad. No obstante, éstos ayudan a fijar los criterios a seguir, las responsabilidades, etc. de todas las personas implicadas en el proceso de producción respecto de la calidad. Es decir, si las especificaciones están bien establecidas, su cumplimiento es la vía para el logro de la satisfacción.

La no conformidad o la existencia de deficiencias en el producto o el proceso, provocan la *insatisfacción* del cliente, expresada en forma de quejas, o de pérdida o deterioro de la relación.

Sin embargo, la satisfacción y la no conformidad no tienen por qué ser conceptos opuestos. Se puede dar el caso, por ejemplo, de un producto que no posea defectos (por tanto, es *conforme*) pero que no se venda, porque

exista otra marca en el mercado que proporcione mayor *satisfacción*.

1.1.3. Planificación, control y mejora de la calidad

La *ingeniería de la calidad* es el conjunto de actividades que una compañía lleva a cabo, tanto en el plano del diseño como en el operativo y el directivo, con el fin de asegurar que las características de calidad de un determinado producto posean los niveles requeridos.

En el proceso de *pensar, organizar y dirigir* la búsqueda de la calidad, pueden identificarse tres etapas (aplicables también a otros ámbitos empresariales, financieros, etc.): *planificación, control y mejora* (que recuerdan al conocido como ciclo PDCA: *plan, do, check, act*).

Planificación de la calidad. En el proceso de planificación, los responsables del producto identifican quiénes y cómo son sus clientes, y cuáles son sus necesidades a cubrir. Posteriormente, determinan qué características o *requerimientos* debe tener el producto de manera que pueda dar respuesta a dichas necesidades, desarrollan procesos capaces de producir dichas características, y trasladan estos planes al personal operativo.

Control de la calidad. El control de la calidad puede definirse como el conjunto de técnicas y actividades dirigidas a verificar el cumplimiento de los requerimientos de calidad establecidos para el producto. Durante la etapa de control, se llevan a cabo tres tareas, funcionando como un bucle que se retroalimenta:

1. La valoración de los resultados operativos reales.
2. La comparación de los resultados con la norma o el objetivo.
3. La realización de acciones como consecuencia de las diferencias observadas.

Cuanto más rápida sea la respuesta ante una desviación, más uniforme será la calidad del producto.

Mejora de la calidad. La mejora de la calidad consiste en la toma de medidas encaminadas a alcanzar unas cotas de conformidad o de ausencia de defectos significativamente superiores a las conseguidas con anterioridad. Dicho de otro modo, la mejora de la calidad trata de la *reducción de la variabilidad* en los procesos y en los productos.

1.1.4. Control estadístico de la calidad

Tanto en el proceso de *control* como de *mejora*, es habitual el empleo de herramientas de tipo estadístico. Llamamos *control estadístico de la calidad* a la aplicación de técnicas estadísticas para medir y mejorar la calidad de los procesos.

Las técnicas comúnmente utilizadas en el control estadístico de la calidad pueden agruparse en las siguientes tres grandes áreas:

- El muestreo de aceptación.
- El *control estadístico de procesos*, que será detallado a continuación.
- El diseño de experimentos.

Cabe decir que dentro de esta clasificación se incluyen tanto técnicas clásicas como utilidades de más reciente aparición. Concretamente, dentro del control estadístico de procesos, podemos encontrar aportaciones muy interesantes desde el ámbito de la inteligencia artificial (véanse, por ejemplo, [Zorriassatine y Tannock, 1998](#) y [Guh et al., 1999](#)), así como aplicaciones que implican el uso de herramientas de optimización (véanse [Chung, 1991](#), [Al-Sultan y Rahim, 1997](#), [Celano y Fichera, 1999](#), [Zhou y Zhao, 2002](#), [Kethler y Peters, 2004](#), [García-Díaz y Aparisi, 2005](#), [Zhang et al., 2008](#), [Aparisi y de Luna, 2009](#) y [Young, 2010](#), entre otros). Todas estas técnicas están enfocadas a hacer posible el objetivo principal de la ingeniería de la calidad, que no es otro que la *reducción sistemática de la variabilidad*.

Las técnicas de muestreo de aceptación (tanto sobre la materia prima o los recursos como sobre el producto final) son más utilizadas en

etapas tempranas del proceso de producción, en las que las unidades no conformes suelen representar un relativamente alto porcentaje de la producción. Posteriormente, se introducen técnicas de control estadístico de procesos para estabilizar el proceso y reducir su variabilidad. En una etapa más madura del proceso, se hace presente además el uso de herramientas de diseño de experimentos, para analizar los factores que influyen en el comportamiento de la característica de calidad de interés, consiguiendo reducir todavía más la variabilidad del proceso y obteniendo así un producto tan competitivo como sea posible.

En la sección siguiente ampliamos la descripción de las técnicas propias del control estadístico de procesos.

1.2. Control estadístico de procesos

El *control estadístico de procesos*, una subárea del control estadístico de la calidad, se define como la aplicación de técnicas estadísticas para la medición y análisis de las variaciones de los procesos, con el fin de conseguir procesos más estables. Dicho de otra forma, el objetivo del control estadístico de procesos (Ferrer-Riquelme, 2009) es monitorizar el rendimiento de un proceso a lo largo del tiempo para comprobar si el proceso se comporta como es esperado y detectar cualquier evento inusual que pudiera suceder.

1.2.1. Variabilidad, estabilidad y control

Previamente en este capítulo hemos mencionado la *variabilidad* como elemento muy relacionado con la calidad o, más concretamente, con la ausencia de ésta.

La existencia de variabilidad en la manufactura de un producto es inevitable, ya que, en general, no existen dos unidades que sean completamente iguales. La variabilidad es objeto y causa de preocupación en la medida en que las diferencias observadas entre las diversas unidades producidas repercute en la satisfacción del cliente. Por esta razón, el

control de la calidad persigue, entre otras cosas, estudiar, controlar y reducir la variabilidad del proceso.

La estadística juega un papel central en dicha labor, ya que proporciona las herramientas adecuadas para poder manejar la variabilidad de un modo cuantitativo. Por ello, en este contexto las características de calidad, objeto del control estadístico de procesos, son modelizadas y tratadas como variables aleatorias.

Habitualmente, se distingue entre dos fuentes o tipos de variabilidad en un proceso: la variabilidad debida a *causas comunes*, fortuitas o aleatorias (y por tanto, difíciles de eliminar), que se considera inherente al proceso, y la debida a *causas asignables* o especiales.

Las causas comunes de variabilidad, de carácter permanente, dan lugar, en su efecto conjunto, a una pauta de variabilidad estable o predecible, mientras que las causas asignables, que tienen un carácter esporádico o puntual, están asociadas a una variabilidad errática o impredecible.

Es especialmente la variabilidad originada por causas asignables la que puede conducir a resultados inaceptables en el producto final. Uno de los principales objetivos del control estadístico de procesos y, en concreto, de los gráficos de control, es diferenciar entre las variaciones en las características de calidad debidas a causas comunes y las debidas a causas asignables, de manera que permiten prevenir tanto los ajustes innecesarios del proceso como la falta de reacción ante variaciones relevantes.

Se dice que un proceso está *bajo control estadístico* cuando no presenta variaciones debidas a causas asignables o, equivalentemente, cuando la distribución de probabilidad de la característica de calidad es constante en el tiempo; en ese caso, también se dice que el proceso es o está *estable*. Por contra, un proceso que opera en presencia de causas asignables se dice que está *fuera de control*. Cuando, por medio del control estadístico de procesos, se identifican y eliminan las posibles causas asignables o especiales de variabilidad, se dice que se ha conseguido *estabilizar* el proceso.

Como herramienta propia del control estadístico de procesos, los grá-

ficos de control, a los que dedicamos la subsección 1.2.2, están orientados a estabilizar un proceso, es decir, a detectar y eliminar la variabilidad debida a causas asignables. En la subsección 1.2.3 abordaremos un análisis complementario, centrado principalmente en la variabilidad inherente o propia del proceso y en el cumplimiento de las especificaciones.

1.2.2. Gráficos de control

De entre los diferentes métodos que engloba el control estadístico de procesos (algunos tan sencillos pero útiles como el histograma, el diagrama de tallo y hojas o el diagrama de Pareto), destacan como herramienta estrella, ampliamente conocida y usada, los *gráficos de control*.

Introducidos por [Shewhart \(1931\)](#) en la primera mitad del siglo XX, los gráficos de control constituyen, todavía hoy, una herramienta básica en la monitorización de un proceso, muy útil para detectar comportamientos inusuales del mismo.

[Montgomery \(2012\)](#) proporciona una descripción detallada y completa de los gráficos de control clásicos y de la mayoría de gráficos *avanzados* que se presentan en los apartados siguientes.

Introducción a los gráficos de control. Gráfico \bar{X} -R

Un gráfico de control consiste básicamente en representar en un diagrama la evolución en el tiempo de una o varias medidas o *estadísticos* derivados a partir de muestras del proceso que son tomadas periódicamente. El gráfico contiene una *línea central* (*central line*, CL), que representa el valor en torno al cual varía en condiciones normales la medida graficada, y, habitualmente, dos líneas más, una a cada lado de la línea central, cuyos valores sirven para delimitar la zona en que, estadísticamente, la gran mayoría de los valores calculados deberían situarse si el proceso opera bajo control (es decir, en presencia sólo de causas comunes de variabilidad), de tal modo que el hecho de que un valor se grafique fuera de dicha zona se interpreta como una señal fundamental de *falta de control* (es decir, un síntoma de que el proceso puede estar fuera de control, ya

que de lo contrario sería muy improbable obtener tal resultado). Dichas líneas reciben el nombre de *límite de control superior* (*upper control limit*, UCL) y *límite de control inferior* (*lower control limit*, LCL).

Adicionalmente, se definen otras configuraciones que, en caso de producirse, también se interpretan como señales de falta de control, y que están basadas en el comportamiento conjunto de varias muestras consecutivas; por ejemplo: obtener 7 u 8 puntos consecutivos por encima o por debajo de la línea central, u observar una *racha* creciente o decreciente de al menos 6 puntos. Estas reglas complementarias, introducidas originalmente por la [Western Electric Company \(1956\)](#), tienen como objeto la detección temprana de situaciones de falta de control. Como inconveniente, incorporar muchas señales adicionales de falta de control provoca el incremento de la frecuencia de *falsas alarmas*, es decir, la aparición de dichas señales con demasiada facilidad aun cuando el proceso se encuentra realmente bajo control estadístico.

El llamado *gráfico \bar{X} -R*, uno de los más utilizados, toma muestras de tamaño constante n a intervalos regulares, y para cada una de ellas calcula y traslada a sendos gráficos, dibujados en paralelo, los estadísticos \bar{x} y R (media y rango, respectivamente), donde

$$\bar{x} = \frac{x_1 + \cdots + x_n}{n} \quad \text{y}$$

$$R = x_{\text{máx}} - x_{\text{mín}},$$

siendo $\{x_1, \dots, x_n\}$ el conjunto de n observaciones consecutivas de la característica de calidad X que interesa controlar, y que en el caso de este gráfico sería una característica numérica continua. En control de la calidad, a este tipo de características se las llama *variables*, en contraposición a las características discretas (conteo de defectos, conteo de unidades defectuosas, etc.), que son llamadas *atributos*.

El *doble gráfico \bar{X} -R* se utiliza para monitorizar simultáneamente la media y la variabilidad de la característica de calidad objeto de interés. Los límites de control para los respectivos estadísticos, \bar{x} y R , son calculados a partir de las propiedades estadísticas conocidas de los

mismos, asumiendo la normalidad de la variable subyacente X . La figura 1.1 muestra un ejemplo de este tipo de gráfico.

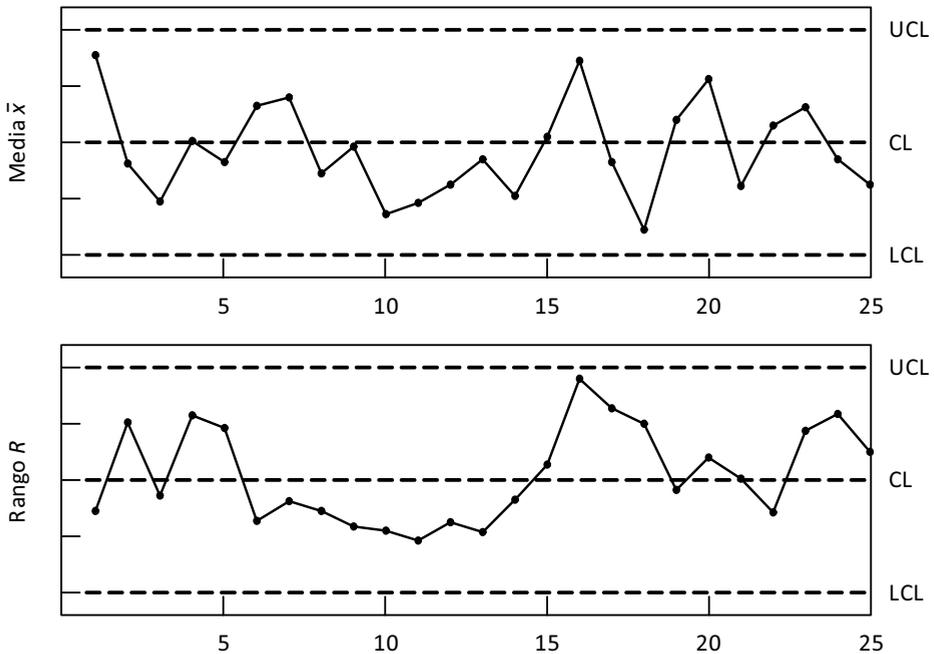


Figura 1.1: Ejemplo de gráfico de control \bar{X} - R . Elaboración propia.

Otros gráficos tradicionales para la monitorización de variables

Otros gráficos clásicos para el control de *variables* son el *gráfico \bar{X} - S* (que es tradicionalmente preferido al \bar{X} - R cuando $n > 10$ —aunque se recomienda $n \geq 20$ — o bien es variable), el *gráfico de medianas* (que, si bien es menos eficiente que el \bar{X} - R , en ocasiones puede ser recomendable por su sencillez y claridad) y el *gráfico de Shewhart para medidas individuales* (de aplicación en situaciones en las que, por las características del proceso, el tamaño muestral n se limita a 1 observación).

Gráficos tradicionales para la monitorización de atributos

Los gráficos clásicos de control para *atributos* son los *gráficos P* y *NP* (para el control del número de unidades defectuosas; grafican, respectivamente, la proporción observada \hat{p} de unidades defectuosas, y la cantidad de unidades defectuosas $n\hat{p}$ en cada muestra), y los *gráficos U* y *C* (para el control del número de defectos en la muestra —donde cada unidad puede, teóricamente, presentar más de un defecto—; representan, respectivamente, la tasa y la cantidad de defectos). En el caso de los gráficos *P* y *U*, las muestras que dan lugar a cada punto del gráfico no tienen por qué ser necesariamente del mismo tamaño.

En el enfoque clásico de [Shewhart](#), los límites de control se calculan asumiendo un modelo probabilístico binomial, para el caso de los gráficos *P* y *NP*, y un proceso de Poisson para los gráficos *U* y *C*, aplicando en todos los casos aproximaciones por la distribución normal.

[Woodall \(1997\)](#) realiza una exhaustiva revisión de los gráficos para el control de atributos, incluyendo tanto tradicionales como avanzados.

Gráficos de valores acumulados y ponderados

Los gráficos de *sumas acumuladas* (*cumulative sum*; CUSUM) y de *medias móviles ponderadas exponencialmente* (*exponentially weighthed moving average*; EWMA) surgieron posteriormente y mejoran el comportamiento de los gráficos de Shewhart en etapas en que el proceso está operando bajo condiciones estables y conocidas tras haber detectado y eliminado las causas especiales de variabilidad más importantes y en las que, por tanto, se espera que, en caso de producirse, las desviaciones del proceso sean relativamente pequeñas. Los gráficos de Shewhart son poco sensibles a tales pequeños cambios, así como a cambios lentos y progresivos, debido a que sólo tienen en cuenta la información de la última muestra graficada para definir una señal fundamental de falta de control. A fin de evitar esto, los gráficos CUSUM y EWMA acumulan la información de las muestras anteriores en cada punto representado.

En concreto, el gráfico CUSUM ([Page, 1954, 1961](#); [Hawkins, 1993](#);

Hawkins y Olwell, 1998) representa la suma de las desviaciones de cada punto que se graficaría en el gráfico de Shewhart correspondiente (la media \bar{x} en el caso del gráfico \bar{X} , por ejemplo) respecto del valor objetivo. Es decir, en el gráfico CUSUM para la media del proceso, lo que se representa para la muestra j -ésima es el valor $C_j = \sum_{h=1}^j (\bar{x}_h - \mu_0) = C_{j-1} + (\bar{x}_j - \mu_0)$, siendo μ_0 el valor objetivo y esperado de \bar{X} . Obviamente, el valor central del gráfico alrededor del cual deberían situarse los valores calculados C_j en el caso de que el proceso opere en ausencia de causas asignables de variabilidad es el cero. Puede construirse una *versión CUSUM* de la mayor parte de los gráficos tradicionales (\bar{X} , R , S , NP , C).

Por su parte, el gráfico EWMA (Roberts, 1959; Hunter, 1986; Crowder, 1987, 1989; Lucas y Saccucci, 1990) incorpora en cada punto representado z_j información de las muestras anteriores a través de la expresión $z_j = \lambda \bar{x}_j + (1 - \lambda)z_{j-1}$, con $z_0 = \mu_0$ y siendo λ un parámetro constante tal que $0 < \lambda \leq 1$; esto en el caso del gráfico de control para \bar{X} , y de manera similar para otros parámetros objeto de monitorización.

Ambos gráficos, CUSUM y EWMA —sobre todo este último—, son utilizados frecuentemente sobre medidas individuales (muestras de tamaño $n = 1$).

Gráficos para ciclos de producción cortos

Existen variantes de los gráficos clásicos para variables y atributos adaptados al caso de sistemas con series de producción cortas (en inglés, *short production runs*), en las que no es posible estimar, por falta de muestras suficientes, los parámetros del proceso ni establecer límites de control válidos a partir de dichas estimaciones de la manera en que requiere un gráfico de control tradicional.

La más sencilla de estas técnicas calcula y representa la desviación respecto del valor objetivo o *nominal*, y por ello suele ser conocida por *gráfico DNOM* (Cullen y Bothe, 1989). En el caso de que no se pueda asegurar que la desviación típica del proceso se mantiene constante entre las diferentes extracciones de muestras, se propone usar el llamado *gráfico*

\bar{X} - R estandarizado (Montgomery, 2012). Una propuesta de generalización de este último es debida a Farnum (1992).

Los gráficos Q propuestos por Quesenberry (1991c,a,b) a partir de una idea de Hawkins (1987) son otra alternativa para ejecuciones cortas basada en estandarizaciones diferentes a la de Farnum. Del Castillo y Montgomery (1994, 1996) aportan diversas mejoras a los gráficos Q y otras alternativas para series cortas. Posteriormente Quesenberry (1995d,b,a,c) presenta también algunas modificaciones que mejoran el comportamiento de los gráficos Q .

Un gráfico de control \bar{X} - R para lotes de tamaño pequeño es desarrollado por Pillet (1996). El gráfico utiliza la información de cada observación, recalculando los límites de control a medida que se muestrean nuevas piezas, de modo que puede generarse una señal de falta de control incluso antes de que se complete una (sub)muestra.

Otras propuestas relativamente recientes son el gráfico de control de Tukey de Alemi (2004) (que se basa en los cuartiles y en el rango intercuartílico para definir los límites de control), y el gráfico t de Zhang et al. (2009) (basado en el estadístico t de Student); ambos presentan un buen comportamiento en procesos con ciclos de producción cortos, según Torng et al. (2009) y Celano et al. (2011), respectivamente.

Asimismo, es posible adaptar los gráficos CUSUM y EWMA a este tipo de ciclos. Como después se destacará en las secciones 1.3 y 2.1, el precontrol es una herramienta de control de procesos que también resulta de utilidad en series cortas.

Gráficos de control adaptables

Como se ha indicado anteriormente, los gráficos de control tradicionales se basan, por lo general, en datos procedentes de muestreos de tamaño y frecuencia constantes (lo que se conoce como *fixed sample size and sampling interval*, FSSI). Posteriormente se ha investigado la posibilidad de adaptar dichos parámetros del procedimiento de control en función de los resultados que se van obteniendo en los sucesivos muestreos que forman parte del mismo, dando lugar a alternativas que mejoran el

comportamiento de los gráficos FSSI. Estas variantes son conocidas de manera global como *gráficos de control adaptables*.

En concreto, Reynolds et al. (1988) exploraron la posibilidad de variar el intervalo de muestreo, dando lugar a los *gráficos VSI* (por *variable sampling interval*), que fueron posteriormente estudiados por Runger y Montgomery (1993), Amin y Miller (1993), Reynolds et al. (1996), Reynolds (1996a) y Reynolds (1996b).

La alternativa de flexibilizar el tamaño muestral de acuerdo con lo que se observa durante los sucesivos muestreos fue propuesta por Prabhu et al. (1993) y Costa (1994), lo cual se conoce como *gráficos VSS* (por *variable sample size*). Aparisi (1996) presenta, por ejemplo, una versión VSS del gráfico T^2 , al que nos referiremos más adelante.

Por último, Prabhu et al. (1994), Prabhu et al. (1997), Costa (1997) y Das y Jain (1997) estudian el diseño de gráficos para el control de la media con tamaño y frecuencia de muestreo variables o *gráficos VSSI* (por *variable sample size and sampling interval*). Costa (1999) desarrolla también una versión VSSI del gráfico \bar{X} -R. Se ha probado que los gráficos VSSI para \bar{X} mejoran el comportamiento del correspondiente gráfico tradicional FSSI en presencia de una (Prabhu et al., 1994) o varias (Lee et al., 2007) causas especiales de variabilidad.

En la misma línea, Carot et al. (2002) presentan una variante del gráfico \bar{X} basada en doble muestreo con intervalo de muestreo variable (DSVSI, por *double sampling variable sampling interval*) y demuestran que ésta consigue detectar cambios pequeños y moderados en la media del proceso más rápido que los correspondientes gráficos VSI, VSS, VSSI, CUSUM y EWMA.

Dos revisiones reseñables sobre el tema son las publicadas por Tagaras (1998) y Zimmer et al. (2000). En la segunda, los autores concluyen que, en general, se obtienen mejores resultados más rápidamente adaptando el tamaño muestral antes que la frecuencia de muestreo.

Gráficos que tienen en cuenta las especificaciones

En procesos cuya variabilidad natural es pequeña en relación a las *especificaciones* o requisitos del producto realizados por el cliente o el responsable del diseño del mismo (lo que suele conocerse como procesos *altamente capaces*; son conceptos que se introducirán formalmente más adelante, en la subsección 1.2.3), tiene sentido que desviaciones relativamente pequeñas de la media real μ respecto de su valor objetivo μ_0 puedan ser consideradas como no importantes, mientras ello no lleve aparejado un excesivo incremento de la proporción esperada de piezas que incumplirían dichas especificaciones.

En los casos en que, bien por este motivo o por otros similares, es más importante tener en cuenta los límites marcados por las especificaciones que los que se derivan estadísticamente de la variabilidad natural del proceso, son de aplicación los *gráficos de control modificados* (véase Hill, 1956) y los *gráficos de control de aceptación* (Freund, 1957). Ambos están diseñados para mantener dentro de unos valores aceptables el porcentaje estimado de *no conformidades* (incumplimientos de las especificaciones) a través de una definición adecuada de los límites de control para la característica de calidad objeto de las especificaciones.

Como se verá más adelante, el precontrol, al que ya nos hemos referido anteriormente, es también una técnica centrada en las especificaciones del producto. En la subsección 2.1.4 se presenta una comparativa entre el precontrol y los gráficos de aceptación.

Gráficos para sistemas de alto rendimiento

En los gráficos diseñados para procesos *altamente capaces* que terminamos de exponer, el estadístico que se monitoriza es de tipo continuo (frecuentemente la media \bar{X}). Se han desarrollado también gráficos de control específicos para el conteo de unidades defectuosas o de defectos (característica discreta) en ese mismo contexto de bajas tasas de piezas defectuosas o *no conformes* (del orden de partes por millón), es decir, procesos con un alto rendimiento (en inglés, *high-yield processes* o también

near-zero defect environments). En estos casos los gráficos tradicionales de Shewhart para atributos no resultan apropiados (Chan et al., 2002; Montgomery, 2012).

Las alternativas principales son monitorizar el número de unidades correctas extraídas antes de muestrear la próxima defectuosa (lo cual daría pie a una distribución geométrica), o bien el *tiempo* (o, equivalentemente, el número de extracciones, si el ritmo de extracción es constante) entre dos defectos consecutivos (lo que podría modelizarse como una variable aleatoria exponencial). El primer enfoque da lugar a los *gráficos CCC* (por *cumulative count of conforming*; Goh, 1987) y el segundo a los *gráficos CQC* (por *cumulative quantity of conforming*; Chan et al., 2000), también llamados *gráficos exponenciales* (Zhang et al., 2005).

Otra propuesta destacable para la monitorización de procesos con baja tasa de unidades defectuosas es el *gráfico CE* de Zúnica Ramajo et al. (1997), basado en la distribución binomial negativa, retomado después con el nombre de *gráfico BN* por Quaglino (2003), quien propone además algunas mejoras.

Gráficos de control por zonas

Una alternativa a registrar y representar gráficamente el valor concreto resultante de cada muestreo es clasificar dicho valor según su distancia a la línea central del gráfico. Basándose en esta idea, Roberts (1966) propone los *gráficos de suma de rachas* (en inglés, *sum run*), llamados así porque, al igual que los gráficos CUSUM y EWMA, tienen en cuenta en cada punto cierta información acumulada de muestreos anteriores.

Concretamente, en los gráficos de suma de rachas el intervalo desde la CL hasta el UCL es dividido en varios tramos o *zonas*, asignándole a cada zona una puntuación no negativa y creciente según su distancia a la CL; y de manera análoga con el intervalo que va desde la CL al LCL. El gráfico comienza con una puntuación acumulada de cero, la cual se va actualizando según la zona en la que se sitúa cada nueva muestra. Si una muestra dada se sitúa en el lado de la CL opuesto a la muestra anterior, la puntuación acumulada hasta ese momento se desestima, reiniciándola

con el valor correspondiente a la muestra actual. Si en algún momento la puntuación acumulada supera un cierto valor prefijado de antemano, se produce una señal de alarma; en tal caso, una vez revisado el proceso y reanudado el control, la puntuación también se reiniciará con la siguiente muestra.

Estos gráficos fueron posteriormente estudiados por Reynolds (1971) y Champ y Rigdon (1997), así como por Jaehn (1987, 1991), quien propone como caso particular el llamado *gráfico de control por zonas* (*zone control chart*). En el gráfico de control por zonas referido a \bar{X} , los LCL y UCL se sitúan ambos a 3 desviaciones típicas de distancia de la CL, como es habitual. Se definen tres zonas de igual longitud desde la CL hasta el UCL, y una cuarta zona que comprende los valores superiores al UCL. La puntuación que normalmente se asigna a estas zonas es de 0, 2, 4 y 8, respectivamente; y de manera análoga para la parte inferior a la CL. El valor 8 es tomado como el umbral que desencadena una señal de alarma.

Los gráficos de control por zonas han sido estudiados, entre otros, por Davis et al. (1990) y Jin y Davis (1991). Champ y Rigdon (1997) demuestran que, utilizando un número suficiente de intervalos, los gráficos de suma de rachas pueden llegar a ser competitivos respecto a los CUSUM y EWMA.

La idea de clasificar las observaciones según su distancia respecto al valor objetivo es también compartida por el precontrol, como se verá más adelante. El objetivo común que hay detrás de transformar valores originales en algún tipo de puntuación o clasificación es simplificar la medición, el registro y la interpretación de los datos por parte de los operarios.

Gráficos de control multivariantes

Todo lo anterior está referido a características de calidad *unidimensionales*. Existen también herramientas y gráficos de control específicos para el caso en que se desee monitorizar el comportamiento conjunto de una característica de calidad *multidimensional*; realizar un control sobre cada una de las dimensiones por separado puede conducir a tasas de errores

tipo I y II inasumibles (Montgomery, 2012; Kruger y Xie, 2012).

La herramienta de control multivariante más conocida es el gráfico T^2 de Hotelling (1947), diseñado para monitorizar el vector de medias de una variable aleatoria normal multivariante, y que supone una extensión del gráfico \bar{X} al caso multidimensional.

Asimismo, existen versiones multivariantes de los gráficos CUSUM (Crosier, 1988; Pignatiello Jr. y Runger, 1990) y EWMA (Lowry et al., 1992), extensiones lógicas del enfoque para el caso unidimensional.

Para el control de la variabilidad de una característica aleatoria normal multivariante, concretada en su matriz de varianzas-covarianzas, Alt (1985) propone dos gráficos, uno de los cuales es la extensión al caso multidimensional del gráfico S^2 y el otro está basado en la monitorización del determinante de la matriz de varianzas-covarianzas muestral.

Topalidou y Psarakis (2009) realizan una revisión exhaustiva de la literatura relativa a gráficos de control multivariante, tanto para variables como para atributos.

Además de lo expuesto, para el control multivariante de características de calidad de elevada dimensión y/o complejidad se recomienda aplicar previamente técnicas de reducción de la dimensión y de detección de estructuras latentes, tales como el *análisis de componentes principales* (*principal component analysis*; PCA) y la *regresión de mínimos cuadrados parciales* (*partial least-squares regression*; PLS regression). Una completa puesta al día sobre el tema puede encontrarse en Kruger y Xie (2012).

Gráficos de control versus contrastes de hipótesis

Existe cierto paralelismo entre el funcionamiento de los gráficos de control y el de un contraste de hipótesis. Cuando un gráfico de control sobre, por ejemplo, la media \bar{X} de un proceso determina que el proceso está fuera de control porque una muestra ha sobrepasado los límites de control, se está asumiendo, con cierta probabilidad de error, que la media real del proceso μ ha cambiado respecto al valor objetivo o deseado μ_0 (debido a la presencia de alguna causa especial); es decir, el gráfico está actuando como un contraste de hipótesis sobre la media poblacional del

proceso (es la visión de [Box y Kramer, 1992](#), entre otros).

Existen autores que ponen en duda la validez de la interpretación de los gráficos de control como contrastes de hipótesis (por ejemplo, [Hoerl y Palm, 1992](#); [Nelson, 1999](#)). La principal objeción es que, en la práctica, es muy probable que no se dé la necesaria condición de independencia de las observaciones dentro de cada muestra y/o entre muestras. Es decir, en la realidad de un proceso productivo es razonable pensar que pueda existir cierta (auto)correlación entre observaciones consecutivas, con lo cual, en rigor, éstas no podrían ser consideradas *realizaciones independientes* de una cierta variable aleatoria X . De hecho, existen técnicas de control estadístico de procesos específicas para datos autocorrelacionados. En un interesante artículo, [Woodall \(2000\)](#) aborda este y otros controvertidos asuntos que enfrentan la teoría y la práctica del control estadístico de procesos.

No obstante lo dicho, es bastante frecuente en control estadístico de procesos emplear la terminología propia de los contrastes de hipótesis, tales como, por ejemplo, *error de tipo I* y *error de tipo II* para referirse, respectivamente, al hecho de rechazar que un proceso está bajo control cuando realmente lo está y al hecho de no detectar que existe una cierta desviación del proceso. De igual forma, ciertas herramientas comunes en inferencia estadística tales como la curva de potencia o la curva característica operativa (gráficas que representan, respectivamente, las probabilidades de sí/no detectar que el proceso está fuera de control en función de la desviación real del mismo) son también usadas para comparar de manera teórica el comportamiento de dos o más gráficos o herramientas de control entre sí.

1.2.3. Análisis de la capacidad de un proceso

Habitualmente, además de saber si un proceso se encuentra bajo control estadístico, se necesita conocer si el resultado final es o no satisfactorio. Es más, un proceso puede estar estabilizado, y aun así producir un elevado número de piezas que incumplan los requerimientos del cliente, si la variabilidad inherente al proceso es relativamente alta o

si el proceso está descentrado.

En este apartado presentamos un análisis sobre la variabilidad del proceso desde esa perspectiva, es decir, relacionándola directamente con las especificaciones del producto.

Capacidad de un proceso

La *capacidad* de un proceso es la habilidad del proceso para cumplir unas especificaciones dadas; indirectamente, proporciona una medida de la variabilidad de la característica o características de calidad que se consideran de interés. Haciendo un paralelismo con las dos clases de fuentes de variación de un proceso —comunes y asignables—, [Montgomery \(2012\)](#) apunta que esta variabilidad también puede ser considerada desde dos puntos de vista diferentes: la variabilidad instantánea (en un momento del tiempo dado), y la variabilidad en el tiempo.

Dado un proceso de producción donde la característica de calidad que se quiere controlar sigue una distribución normal de media μ y desviación típica σ , es habitual definir la variabilidad *natural* del proceso como seis veces el valor de la desviación típica. En este sentido, se acostumbra a decir también que los límites inferior y superior de tolerancia naturales del proceso son, respectivamente, $\mu - 3\sigma$ y $\mu + 3\sigma$.

Una medida usada frecuentemente y que relaciona los conceptos de variabilidad natural y cumplimiento de especificaciones es la llamada *capacidad potencial* del proceso, C_p , también conocida como relación de capacidad, y que se define de la manera siguiente:

$$C_p = \frac{USL - LSL}{6\sigma}, \quad (1.1)$$

donde USL y LSL son los límites superior e inferior de tolerancia o especificación, respectivamente. Obviamente², supondremos

$$LSL < USL. \quad (1.2)$$

²El caso en que $LSL = USL$ no tiene sentido en la práctica.

El intervalo $[LSL, USL]$ se conoce como *rango de tolerancias*. Cuando un valor observado de la característica de calidad pertenece al rango de tolerancias, se dice también que está dentro de tolerancias o dentro de especificaciones.

La expresión (1.1) se aplica cuando existen especificaciones bilaterales.

En general, podemos estimar σ —y, por tanto, la capacidad potencial C_p — por los medios habituales en estadística. En concreto, cuando es sabido que un proceso está bajo control estadístico, los gráficos de control resultan de utilidad para obtener una estimación $\hat{\sigma}$ de la desviación típica y, a partir de ella, una estimación \hat{C}_p de la capacidad potencial del proceso. Una revisión exhaustiva de procedimientos para estimar la desviación típica σ de un proceso puede encontrarse en [Carot Alonso \(1998\)](#).

La figura 1.2 ilustra el hecho de que cuanto menor sea la variabilidad inherente a un proceso, mayor será su capacidad potencial y, en consecuencia, si está centrado, menor proporción de piezas no conformes producirá. De manera tradicional se denominaba proceso *potencialmente justamente capaz* a aquel cuya capacidad potencial es igual a 1 (es decir, $6\sigma = USL - LSL$), dado que, si es normal y centrado, generaría piezas fuera de especificaciones en una proporción igual a 0,27% (2700 piezas no conformes por cada millón de unidades producidas). En la actualidad, sin embargo, se considera como aceptable, en términos generales, una capacidad potencial no inferior a 1,3333 ($8\sigma \leq USL - LSL$; 63 piezas no conformes por cada millón producido, en un proceso normal y centrado), siendo deseable que alcance valores iguales o superiores a 1,6667 ($10\sigma \leq USL - LSL$ y 0,5 piezas no conformes por millón, bajo los mismos supuestos).

Capacidad versus estabilidad

Tal y como apunta [Montgomery \(2012\)](#), el análisis de la capacidad tiene que ver más con la *uniformidad* que con la *estabilidad* del proceso.

De acuerdo con la síntesis que realiza [Woodall \(2000\)](#), para una característica de calidad, la mejora de la calidad consiste, la mayoría de las veces, en centrar su distribución de probabilidad en un valor

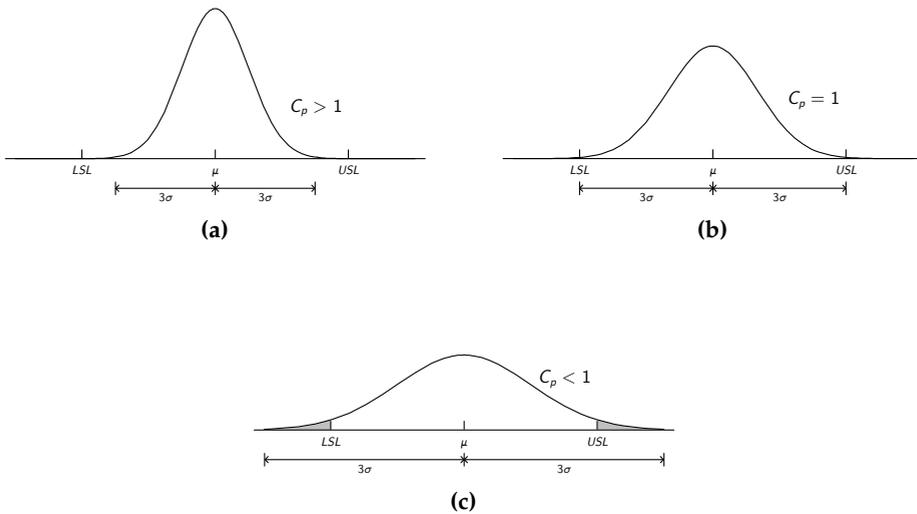


Figura 1.2: Relación de capacidad para tres procesos normales con diferente dispersión. La figura (b) presenta un proceso en el que las tolerancias naturales coinciden con el rango de especificaciones. Las figuras (a) y (c) representan procesos con capacidad potencial alta y baja, respectivamente. Elaboración propia.

objetivo —es decir, conseguir que μ se establezca en un valor deseado o *valor nominal* μ_0 — y en reducir su variabilidad —es decir, reducir σ o, equivalentemente, aumentar C_p —. Según Deming (1986), citado por Woodall (2000), la variabilidad debería reducirse tanto que «las especificaciones se pierdan en el horizonte», de manera que el proceso fabrique piezas mayoritariamente dentro del rango de tolerancias, aun cuando el proceso se encuentre fuera de control. En el mismo artículo, se recoge también la propuesta de Taguchi (1981), más formal y objetiva, consistente en reducir la variabilidad no más allá del punto en que comenzaría a ser económicamente no rentable. Por último, citaremos a Ishikawa (1985), quien destaca como una de las actitudes que deben estar presentes para poder hablar de calidad en la empresa el distinguir entre la «calidad del diseño» y la «calidad en el cumplimiento de los requisitos».

Formalmente, se dice que un proceso está *centrado* si la media real μ de la característica de calidad objeto de interés coincide con el valor

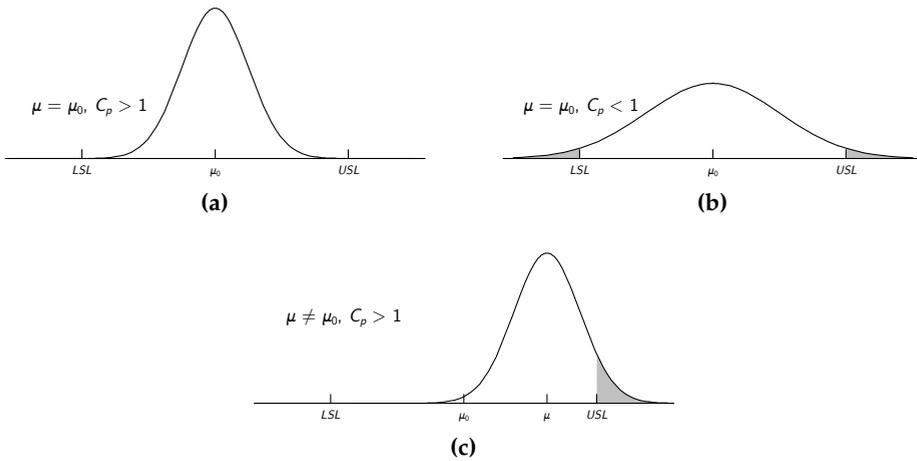


Figura 1.3: Diferencia entre capacidad potencial y real. La figura (a) muestra un proceso con capacidad potencial alta y, además, centrado. Las figuras (b) y (c) representan, respectivamente, un proceso centrado pero con capacidad potencial baja, y un proceso potencialmente capaz pero no centrado; en ambos casos, la proporción de piezas fuera de las especificaciones es apreciable. Elaboración propia.

objetivo o deseado para ella μ_0 , también llamado valor *nominal*, como ya se ha mencionado antes. Asimismo, se dice que el proceso está *descentrado* si $\mu \neq \mu_0$.

Un proceso puede ser estable y estar centrado o no. En la figura 1.3 mostramos algunas situaciones en las que la característica de calidad presenta diferentes funciones de densidad de probabilidad, y lo que ello implica en términos de *capacidad*.

Tal y como se ilustra en la figura 1.3(c), un proceso puede ser *potencialmente capaz* y aun así producir una proporción inaceptable de piezas fuera de especificaciones, a causa de no estar centrado. Un parámetro que suele utilizarse para informar del cumplimiento real, no potencial, de las especificaciones por parte de un proceso es el *índice de capacidad* C_{pk} , que se define de la siguiente forma:

$$C_{pk} = \min \left\{ \frac{USL - \mu}{3\sigma}, \frac{\mu - LSL}{3\sigma} \right\}. \quad (1.3)$$

Como se ve, este índice C_{pk} se calcula como la diferencia entre la media real del proceso μ y el límite de especificación más cercano, puesta en relación con la mitad de la variabilidad natural del proceso. Así pues, este índice relaciona en una única medida la posición (expresada a través de μ) y la variabilidad natural (6σ) del proceso.

Se puede demostrar que, en general, $C_{pk} \leq C_p$, y que si $\mu = \mu_0$ (proceso centrado) y además μ_0 está en el centro del rango de tolerancias (lo cual estamos dando por supuesto en todo momento), entonces y sólo entonces $C_{pk} = C_p$. Por ello, puede decirse que C_{pk} representa una medida real de la capacidad, frente a C_p , que representa la capacidad *en el mejor caso*.

Nótese que, a diferencia de C_p , el índice C_{pk} puede tomar valores negativos. Esto sucede si y sólo si la media real μ se encuentra situada fuera del intervalo $[LSL, USL]$.

En el caso de existir especificaciones unilaterales, superiores o inferiores, las respectivas definiciones para C_{pk} son las siguientes:

$$C_{pk} = \frac{USL - \mu}{3\sigma},$$

$$C_{pk} = \frac{\mu - LSL}{3\sigma}.$$

Para profundizar más en el análisis de la capacidad de un proceso, puede consultarse el capítulo 8 de [Montgomery \(2012\)](#).

1.3. Precontrol

El precontrol es una técnica de control de la calidad que se usa para evaluar la capacidad o aptitud de un proceso de producción para generar unidades que se adecuen a unas especificaciones previamente fijadas. Se caracteriza por su sencillez e inmediatez en el uso (no requiere cálculos, gráficos o anotaciones sofisticados), y por ello suele considerarse idóneo para ejecuciones cortas y valoraciones rápidas. Fue desarrollado en 1952 por un grupo de consultores dirigido por [Satterthwaite \(1954\)](#) y

posteriormente recuperado por [Shainin y Shainin \(1989\)](#).

A pesar de estar diseñados con fines distintos, el precontrol ha sido frecuentemente comparado con los gráficos de control de Shewhart, e incluso se ha planteado como alternativa a éstos (véase [Mackertich, 1990](#), por ejemplo). [Ledolter y Swersey \(1997\)](#), entre otros, evalúan el comportamiento del precontrol como técnica para detectar desviaciones en la media del proceso.

Una de las propuestas más interesantes y novedosas en torno a la técnica clásica es el llamado *precontrol generalizado* ([San Matías et al., 2004](#)), que plantea la mejora de la fase de cualificación del precontrol mediante la flexibilización de ciertos parámetros que determinan el comportamiento de la técnica. Los autores demuestran que eligiendo convenientemente dichos parámetros es posible conseguir un plan de precontrol que se adecue a las exigencias del usuario final en términos de la tasa de falsa alarma y de la potencia para detectar desviaciones del proceso. La decisión del valor a asignar a los parámetros en cada caso concreto queda en manos del usuario.

Esta propuesta muestra un modo de aportar más flexibilidad al precontrol, sin afectar a su sencillez y facilidad de uso. Ahora bien, el usuario tiene a su alcance la posibilidad de elegir entre diferentes configuraciones de la técnica del precontrol, y a priori no está claro si alguna de todas ellas sería más recomendable que las otras. Es decir, *queda pendiente la obtención de la configuración óptima de los parámetros del precontrol*. Esta es la cuestión fundamental a resolver si se pretende sacar rendimiento práctico a esta técnica tan sencilla que a la vez, convenientemente moldeada, puede ser muy potente.

1.4. Optimización en control de la calidad

La aplicación de un enfoque de optimización en el diseño de herramientas de control de procesos es cada vez más frecuente ([Rahim y Al-Sultan, 1997](#)). Las ventajas de aplicar la optimización en la monitorización de procesos, como en cualquier otra área, son claras: obtención de mejores

herramientas de control, adaptadas a las necesidades del usuario, mayor conocimiento del sistema, y ayuda fundamental en el proceso de toma de decisiones.

En particular, el uso de técnicas de optimización para el diseño de herramientas de control de calidad tales como gráficos de control o planes de aceptación de lotes con respecto a criterios económicos ha sido abordado por diversos autores (Tagaras, 1997; Cheng et al., 1997; García-Díaz y Aparisi, 2005), tanto de manera aislada como en conjunto con la planificación de la producción (Goyal et al., 1993) o las políticas de inspección (Rahim y Ben-Daya, 1996), mantenimiento (Rahim et al., 1996), etc. Asimismo, el uso de técnicas de optimización evolutivas tales como algoritmos genéticos no es poco habitual (Chou y Chen, 2006; Vommi y Seetala, 2007).

Una puesta al día y una discusión sobre problemas y técnicas de optimización en control estadístico y mejora de procesos es ofrecida por Carlyle et al. (2000a), Saniga (2000), Del Castillo y Semple (2000), Nelson y Wiecek (2000), Johnson (2000) y Carlyle et al. (2000b).

1.4.1. Programación no lineal entera mixta

De entre las muchas técnicas de optimización existentes, en el caso concreto del presente trabajo, la determinación de la mejor configuración de la técnica del precontrol pasa, como se verá, por la formulación y resolución de un problema de programación no lineal entera mixta, razón por la cual hacemos aquí mención aparte de ella.

La programación no lineal entera mixta, o, en inglés, *Mixed-Integer Nonlinear Programming* (MINLP), es la rama de la programación matemática que se ocupa de los problemas de optimización con presencia de funciones no lineales³ y variables de naturaleza tanto entera como

³Una función $F : \mathbb{R}^p \rightarrow \mathbb{R}$ es *lineal* si y sólo si puede expresarse como producto escalar del vector de variables por un vector de coeficientes constantes (añadiendo, en todo caso, un término independiente), es decir, si existen $\mathbf{c} \in \mathbb{R}^p$ y $c_0 \in \mathbb{R}$ tales que $F(\mathbf{x}) = c_0 + \mathbf{c}\mathbf{x}$ para todo $\mathbf{x} \in \mathbb{R}^p$. Cualquier función que no se ajusta a dicha caracterización se dice que es *no lineal*.

continua. Muchos problemas en ingeniería y en otros ámbitos pueden ser formulados de manera natural como problemas de MINLP (Bussieck y Pruessner, 2003 y Burer y Letchford, 2012, entre otros). Especialmente, los problemas de síntesis en ingeniería química e ingeniería de sistemas de proceso han jugado un papel importante en el desarrollo de las técnicas de MINLP (Grossmann y Sargent, 1979; Yee y Grossmann, 1990; Luyben y Floudas, 1994; Adjiman et al., 1997, 1998; Angira y Babu, 2003; Misener y Floudas, 2009, 2012).

Entre muchos otros, la MINLP se ha aplicado con éxito (Leyffer, 2010; Burer y Letchford, 2012; Belotti et al., 2013) a problemas de ingeniería como el diseño óptimo de redes de distribución de agua (Bragalli et al., 2006; Karuppiah y Grossmann, 2006), de gas (Martin et al., 2006), de energía (Murray y Shanbhag, 2006), de transporte (Fügenschuh et al., 2010) o de telecomunicación (Boorstyn y Frank, 1977); y a diversos casos de ingeniería informática como, por ejemplo, la optimización del rendimiento de redes de fibra óptica (Elwalid et al., 2006), el equilibrado del volumen de transmisión en redes inalámbricas malladas (Guo y Huang, 2011), la asignación óptima de ancho de banda inalámbrico (Costa-Montenegro et al., 2007) o la resolución de ciberataques (Altunay et al., 2011). También en otras áreas de la ingeniería y la industria destacan aplicaciones de la MINLP como la minimización del desperdicio de materia prima en la industria papelera (Harjunkoski et al., 1999), el diseño de componentes electrónicos (van de Braak et al., 2004), la recarga de combustible de reactores nucleares (Quist et al., 1999), la prevención de apagones en sistemas de energía eléctrica (Bienstock y Mattia, 2007; Donde et al., 2005), la minimización del impacto ambiental en plantas industriales (Eliceche et al., 2007), la optimización de estructuras de construcción (Guerra et al., 2011), el diseño óptimo de sistemas de aislamiento térmico (Abhishek et al., 2010b), la búsqueda de un sistema sostenible de producción y suministro de combustible (Corsano et al., 2011), el diseño de planes de respuesta ante vertidos de crudo (You y Leyffer, 2011), la optimización del embarque en aviones (van der Briel et al., 2005) o el diseño de la distribución de bloques (departamentos, estancias, etc.) en industrias y

servicios (Castillo et al., 2005).

Otras aplicaciones de la MINLP a destacar en el ámbito de la logística, la economía y la mercadotecnia son la optimización de carteras de inversión (Bienstock, 1996; Jobst et al., 2001), así como otras relacionadas con la planificación de la producción, la localización de infraestructuras y servicios, sistemas de distribución, etc. (véanse los trabajos de Adams y Sherali, 1993 y Freire et al., 2012). También, en medicina es reseñable, por ejemplo, el artículo de Pardalos et al. (2004) en el que se utiliza la MINLP para ajustar modelos predictivos de ataques de epilepsia.

La forma general de un problema de MINLP es la siguiente:

$$\left. \begin{array}{l} \text{Min} \quad f(\mathbf{x}, \mathbf{y}) \\ \text{s.a:} \quad g_j(\mathbf{x}, \mathbf{y}) \leq 0 \text{ para } j \in J \\ \quad \quad \mathbf{x} \in X, \mathbf{y} \in Y \cap \mathbb{Z}^{p_2} \end{array} \right\}, \quad (1.4)$$

siendo f y g_j , $j \in J$, funciones no lineales (todas o algunas de ellas) diferenciables, J el conjunto de índices de las restricciones y \mathbf{x} e \mathbf{y} las variables de decisión continuas y discretas, respectivamente, con $X \subseteq \mathbb{R}^{p_1}$ e $Y \subseteq \mathbb{R}^{p_2}$, $p_1, p_2 \geq 1$; generalmente, X e Y son espacios poliédricos⁴. Recuérdese que \mathbb{R} y \mathbb{Z} representan los conjuntos de los número reales y enteros, respectivamente.

1.4.2. Métodos de resolución de MINLP⁵

Por su doble naturaleza no lineal y entera, los problemas de MINLP no son, en general, sencillos de resolver. De hecho, el problema genérico (1.4) es uno de los más complejos dentro del campo de la Optimización, y

⁴Un subconjunto $X \subseteq \mathbb{R}^p$ es un espacio poliédrico si puede expresarse como intersección de una cantidad finita r de semiespacios de \mathbb{R}^p , es decir, si existen $\mathbf{A} \in \mathbb{R}^{r \times p}$ y $\mathbf{b} \in \mathbb{R}^r$ tales que $X = \{ \mathbf{x} \in \mathbb{R}^p \mid \mathbf{Ax} \leq \mathbf{b} \}$ para algún $r \geq 1$. Algunos autores también consideran espacios poliédricos a la unión (no necesariamente finita) de conjuntos de esta clase.

⁵En esta subsección se utilizan términos propios del área de la Investigación Operativa que se suponen conocidos, tales como el concepto de *problema convexo*, *plano de corte*, *algoritmo heurístico*, etc. Referimos al lector no iniciado a textos básicos tales como los de Hillier y Lieberman (2010) y Maroto Álvarez et al. (2002).

pertenece a la clase de los problemas *NP-completos*⁶ (Leyffer, 1993; Köppe, 2011; Belotti et al., 2013).

Algoritmos clásicos

Los primeros algoritmos específicos para problemas de MINLP surgen en los años 70, basados en una generalización del método de descomposición de Benders (Geoffrion, 1972), y también como extensión de la técnica de *branch-and-bound* al caso no lineal (Beale, 1977), esta última ya sugerida por Dakin (1965).

Posteriormente, se han sucedido en la literatura una serie de propuestas de métodos deterministas o *exactos* para MINLP, que abordan la resolución del problema mediante diferentes estrategias. Se trata de métodos que permiten asegurar la convergencia a un óptimo global bajo ciertas condiciones como, por ejemplo y principalmente, la convexidad o pseudoconvexidad del problema; si dichas condiciones no se cumplen, habitualmente estos métodos sólo pueden garantizar la convergencia a un óptimo local (o ni siquiera eso, en algunos casos). Podemos agruparlos como sigue:

Métodos basados en *branch-and-bound* (abreviado de manera habitual como NLP-B&B o simplemente B&B; Gupta y Ravindran, 1981, 1985; Sandgren, 1990; Bonami et al., 2011). Fundamentan su estrategia de búsqueda en sucesivas subdivisiones del espacio muestral, al estilo del algoritmo de *branch-and-bound* para problemas lineales enteros, con la diferencia de que dentro de cada nodo se resuelve un problema no lineal.

Suelen incluirse también dentro de esta categoría algoritmos tipo *branch-and-cut* (Stubbs y Mehrotra, 1999; Belotti et al., 2013), que,

⁶De manera informal, se puede decir que un problema NP-completo es aquel para el que no se conoce y *muy probablemente* no exista un algoritmo *determinista* que sea capaz de resolverlo *en tiempo polinomial* (entendiendo por ello que el tiempo necesario para resolverlo puede expresarse como una función polinómica del número de variables, *inputs* o grados de libertad del problema). El concepto forma parte de la teoría de la complejidad computacional y fue introducido por Cook (1971).

en esencia, añaden a la estrategia de búsqueda en forma de árbol el uso de planos de corte, eliminando en cada nodo (y en sus descendientes) soluciones fraccionarias sin eliminar ninguna entera, con el objetivo de reducir el tamaño final del árbol.

Otra estrategia utilizada dentro de los algoritmos tipo *branch-and-bound*, con el fin de ahorrar operaciones, es la de detenerse antes de obtener el óptimo en la resolución de los problemas de programación no lineal (NLP, por sus siglas en inglés) que se generan en cada nodo. En esta línea se encuadran los trabajos de Borchers y Mitchell (1994), Leyffer (2001) y más recientemente Mahajan et al. (2012).

Métodos basados en aproximación exterior (*outer approximation*, OA; Duran y Grossmann, 1986; Kocis y Grossmann, 1987; Yuan et al., 1988; Fletcher y Leyffer, 1994; Bonami et al., 2008). Constituyen la otra gran familia de métodos clásicos para la resolución de problemas de MINLP convexos. La estrategia de éstos consiste, en primer lugar, en resolver el problema no lineal resultante de fijar las variables enteras del problema de MINLP a un valor inicial dado; esto, en el caso de que el problema de NLP resuelto sea factible, proporciona una cota (superior si el problema original es de minimizar) de la solución óptima entera. A continuación, esta solución hallada se utiliza como base para construir un modelo de programación lineal entera (*Mixed-Integer Linear Programming*, MILP) resultante de relajar las restricciones no lineales (convexas) del problema de MINLP mediante restricciones lineales (hiperplanos de soporte). Dicho problema se resuelve, proporcionando una cota (inferior si el problema es de minimizar) de la solución óptima del problema original, y el punto óptimo obtenido se utiliza como base para fijar de nuevo las variables enteras y generar un nuevo problema de NLP. Se sucede, por tanto, una resolución alternada iterativa de problemas de NLP y de MILP, proporcionando cada vez mejores cotas, y que suele converger en pocos pasos (Caballero y

Grossmann, 2007).

Descomposición de Benders generalizada (*generalized Benders decomposition*, GBD; Geoffrion, 1972; Van Roy, 1983). Aunque este algoritmo surgió antes que los métodos de OA, se puede considerar una variante de éstos (Belotti et al., 2013). Difiere de la OA en la manera como se construyen las restricciones del problema de MILP resultante de relajar el problema original (entre otras cosas, sólo se consideran las restricciones activas en cada paso). De hecho, Duran y Grossmann (1986) demuestran que la GBD puede ser vista como un caso particular de los métodos de OA, y que las cotas que proporciona en la resolución de cada problema de MILP son siempre peores que las obtenidas por el problema equivalente en OA, motivo por el cual la GBD suele necesitar un mayor número de iteraciones para converger (Caballero y Grossmann, 2007).

Plano de corte extendido (*extended cutting plane*, ECP; Westerlund y Pettersson, 1995; Westerlund y Pörn, 2002; Still y Westerlund, 2009). Es una generalización de la técnica de planos de corte, ideada inicialmente por Kelley Jr (1960) para problemas convexos no lineales continuos. Asimismo, puede ser visto como una variante de la OA en la que *no* se resuelven problemas no lineales, sino sólo de MILP. El problema auxiliar de NLP es sustituido aquí por otro de MILP, resultante de añadir una linealización de la restricción no lineal que haya sido más violada en el paso anterior (en el paso inicial, el problema de MILP incluye sólo las restricciones lineales que pueda haber en el problema original de MINLP). Así, el tamaño del problema de MILP con que se trabaja va creciendo en cada iteración, por lo que se hace necesario contar con métodos de resolución eficientes para dicha clase de problemas (Still y Westerlund, 2009). El método puede requerir un número elevado de iteraciones para converger, aunque en la práctica es competitivo con respecto a los métodos de OA (Belotti et al., 2013).

Nótese que tanto la OA como la GBD y el método del ECP resuelven

en cada iteración un problema de MILP (además de uno de NLP, en el caso de la OA y la GBD), esto es, aplican algoritmos tipo *branch-and-bound* (para el caso lineal, en esta ocasión) u otros similares en cada iteración. Es por esta razón que a veces se los clasifica como *métodos multiarbol* (Belotti et al., 2013).

Branch-and-bound basado en programación lineal y no lineal (abreviado normalmente como LP/NLP-B&B o también QG, por las iniciales de sus autores; Quesada y Grossmann, 1992). Este algoritmo combina una estrategia de búsqueda tipo *branch-and-bound* con ideas de los métodos de OA; puede ser considerado como un híbrido de ambos enfoques. El método se inicia, como en la OA, fijando las variables enteras a un valor arbitrario y resolviendo el problema de NLP resultante. En lugar de generar a partir de la solución hallada un problema de MILP utilizando relajaciones lineales de las restricciones, como se hace en la OA, el algoritmo de QG utiliza dicha solución factible como base para bifurcar la región de búsqueda, al estilo del *branch-and-bound*. Cuando en un nodo se llega a una solución entera (cota superior del óptimo entero si el problema es de minimizar), se vuelve a resolver un problema de NLP fijando las variables enteras al valor hallado, y con la nueva cota que éste proporciona, se actualiza el estado de los nodos del árbol pendientes de explorar. Se puede demostrar que el método termina convergiendo en un número finito de pasos. En la práctica, la resolución de los problemas no lineales en el algoritmo de QG no supone una gran carga computacional y consigue reducir bastante el tamaño del árbol.

Abhishek et al. (2010a) mejoran el rendimiento del algoritmo de QG aplicando técnicas de búsqueda avanzadas de MILP y el uso de planos de corte propios de la técnica del ECP.

Otros enfoques híbridos clásicos. Caballero y Grossmann (2007) y Belotti et al. (2013) apuntan que la hibridación del enfoque de B&B con los métodos multiárbol (cuyo resultado es un algoritmo

de un solo árbol) puede aplicarse no sólo a la OA (método de QG), sino también a la GBD o al método del ECP. Esta última opción es precisamente el fundamento del algoritmo del plano de corte secuencial (*sequential cutting plane*; SCP) propuesto por [Still y Westerlund \(2006\)](#).

Métodos de optimización global

En la literatura existen también propuestas que abordan de forma exacta la búsqueda del óptimo global de problemas de MINLP *no convexos* (aquellos cuya relajación continua involucra funciones no convexas, bien en el objetivo o bien en las restricciones). Obviamente, este caso es mucho más complicado de resolver. De hecho, la llamada Optimización Global, a la que está ligada la MINLP no convexa, es un área de la Investigación Operativa bastante activa en los últimos tiempos, por su interés y complejidad, y con muchas aplicaciones prácticas en la ingeniería y en la ciencia en general (véase, por ejemplo, la revisión de [Floudas y Gounaris, 2009](#)).

La mayoría de los métodos para problemas de MINLP no convexos son extensiones de los métodos clásicos descritos en la subsección anterior. De hecho, un posible enfoque, aplicable sólo a casos muy concretos, consiste en reformular completamente el problema de MINLP no convexo para convertirlo en uno equivalente que sea convexo, y así poder aplicar las técnicas ya descritas. El trabajo de [Liberti et al. \(2009\)](#) profundiza en esta línea. A parte de esto, las estrategias utilizadas para diseñar algoritmos de MINLP para el caso no convexo pueden clasificarse en dos grandes grupos ([Belotti et al., 2013](#)):

Aproximación de las funciones no lineales mediante funciones lineales a tramos. En esencia, esto permite resolver el problema de MINLP original mediante técnicas de MILP. Véanse [Geißler et al. \(2012\)](#) y [Vielma et al. \(2010\)](#) para más detalles.

Uso de relajaciones convexas. Es el recurso más utilizado. Existen diferentes estrategias, tanto genéricas como específicas, para lograr rela-

jar problemas enteros no lineales no convexos mediante funciones convexas. Esto permite recurrir a los métodos para MINLP convexa tratados anteriormente y, mediante procedimientos iterativos que cada vez consigan ajustar más las relajaciones, llegar a una solución óptima.

Una de las aplicaciones directas más conocidas de este último enfoque es el llamado *branch-and-bound* espacial. Es una variante del algoritmo *branch-and-bound* en la que se permite bifurcar por cualquier variable, no sólo por aquellas definidas como enteras (de hecho, es necesario para poder garantizar de manera teórica la convergencia del algoritmo; McCormick, 1976; Horst y Tuy, 1996). Tras bifurcar por una variable dada, se busca una relajación convexa en cada uno de los dos subproblemas hijo (en principio, disjuntos) que se han generado (es decir, la relajación podría no ser convexa considerando el problema padre en su totalidad, pero sí lo es en cada subproblema hijo, lo cual permite conseguir relajaciones cada vez más ajustadas según se va profundizando en el árbol). Algunas implementaciones destacables del *branch-and-bound* espacial pueden encontrarse en Ryoo y Sahinidis (1996), Tawarmalani y Sahinidis (2004) y Belotti et al. (2009).

Asimismo, encontramos en la literatura otras extensiones del B&B al caso no convexo basadas en relajaciones convexas, como por ejemplo el trabajo de Misener y Floudas (2013). Kesavan y Barton (2000) desarrollan un algoritmo tipo *branch-and-cut* para problemas no lineales no convexos. Por otro lado, Still y Westerlund (2009) proponen una generalización de las técnicas de ECP para MINLP no convexa.

Métodos sin uso de derivadas

En problemas de optimización reales no es extraño encontrarse con situaciones en las que la función objetivo y/o las restricciones se comportan como una *caja negra*, lo que significa que *no* es posible obtener una expresión analítica explícita para ellas o, si la hay, su cálculo directo es inviable en términos de coste computacional, de tal modo que, dada

una solución cualquiera, el resultado de la función objetivo para esa solución y/o el hecho de si ésta pertenece o no a la región factible sólo es averiguable mediante un proceso empírico o numérico que habitualmente es muy costoso en tiempo (esto se da a menudo, entre otros campos, en ingeniería química, Meyer et al., 2002; otros casos de uso recientes son, por ejemplo, el diseño de revestimientos de suelos, Bartelt-Hunt et al., 2006; el diseño de nanocomponentes, Maria et al., 2009; o la optimización automática de código informático, Balaprakash et al., 2011). Este tipo de problemas suponen un especial desafío (véase, por ejemplo, Jones et al., 1998), ya que muchas de las técnicas clásicas de optimización requieren conocer expresiones explícitas de las funciones implicadas en el problema a resolver. Egea et al. (2007) mencionan las técnicas más destacadas para problemas de esta clase.

Por otra parte, una clase diferente de problemas que, sin ser necesariamente de tipo *caja negra*, requieren asimismo de algoritmos con enfoques alternativos a los clásicos es aquella en la que se involucran funciones para las cuales no es posible en la práctica calcular sus derivadas, bien porque no sean derivables, o bien porque dicho cálculo suponga un coste computacional inasumible. Para más detalles sobre este tema, puede consultarse el trabajo de Conn et al. (2009).

Dado que ninguno de estos es el caso de nuestro problema, no entraremos aquí a revisar a fondo todos los métodos para problemas de MINLP tipo *caja negra* o sin uso de derivadas, si bien reiteramos que se trata de un área de estudio muy interesante.

Técnicas heurísticas

Como ya se ha mencionado, los problemas de MINLP pueden llegar a ser muy difíciles de resolver de manera óptima en un tiempo razonable. El uso de técnicas de Optimización heurísticas (es decir, no exactas) y metaheurísticas permite abordar dichos casos de manera eficiente, aunque sin garantizar en general la optimalidad.

Nos referimos en este subapartado a las técnicas *propiaamente heurísticas* para MINLP, si bien, como ya hemos destacado anteriormente, cualquier

técnica exacta para MINLP convexa aplicada sobre un problema no convexo también puede ser considerada, en cierto modo, como heurística.

En las últimas décadas han surgido algoritmos heurísticos y metaheurísticos que, mediante toda clase de enfoques —*simulated annealing* (Kincaid, 1992; Cardoso et al., 1997), búsqueda tabú (Lin y Miller, 2004), *scatter search* (Ugray et al., 2005), algoritmos genéticos (Rajeev y Krishnamoorthy, 1992; Young et al., 2007), algoritmos evolutivos en general (Cao y Wu, 1997; Lin et al., 1999; Costa y Oliveira, 2001; Babu y Angira, 2002; Munawar, 2005; Dimopoulos, 2007; He y Wang, 2007; Luo et al., 2007), etc.—, pretenden abordar la búsqueda de soluciones óptimas o, en su defecto, tan buenas como sea posible, para problemas de MINLP más o menos genéricos.

Además de estos algoritmos, que son el resultado de adaptar esquemas heurísticos generales al caso particular de la MINLP, existen propuestas específicamente pensadas para problemas no lineales enteros. Por ejemplo, Nannicini y Belotti (2012) y Berthold (2012) presentan sendos algoritmos heurísticos para MINLP basados en redondeo. Viswanathan y Grossmann (1990) proponen una heurística que extiende la técnica de OA al caso no convexo.

Un algoritmo heurístico de búsqueda, pensado para ser aplicado de manera auxiliar junto con otra técnica de mejora de soluciones, es la llamada *feasibility pump* (FP; *bomba de factibilidad*) de Bonami et al. (2009). La FP también aplica redondeo, pero partiendo de una aproximación del modelo de MINLP original basada en OA.

Asimismo, algunos heurísticos a destacar orientados a la mejora de una solución dada en MINLP son los presentados por Nannicini et al. (2008), Liberti et al. (2011) y Bonami y Gonçalves (2012).

Técnicas auxiliares para la mejora del rendimiento

En MINLP cobra bastante importancia el trabajo que se realice para adaptar el modelo de manera previa a su resolución, así como otras operaciones encaminadas a mejorar el rendimiento o el resultado final del algoritmo principal. Dentro de las técnicas orientadas a este objetivo,

podemos distinguir las siguientes clases:

Técnicas de preproceso. Las técnicas de preproceso fueron popularizadas por Savelsbergh (1994) en MILP, y son extensibles al caso de la MINLP sin muchas dificultades (Belotti et al., 2013). Algunas de estas técnicas consisten en reformular el problema a resolver, mediante, por ejemplo, el endurecimiento de restricciones (hacer más restrictivas algunas condiciones sin eliminar ninguna solución válida) o la desagregación de restricciones (sustituir una restricción dada por varias equivalentes), siempre con el fin de que la región factible correspondiente a la relajación continua del problema no sea mucho más grande que la envoltura convexa de la región factible original, lo cual, en principio, facilita la resolución de los problemas no lineales auxiliares que se contemplan.

Técnicas de búsqueda de soluciones iniciales y de determinación de infactibilidad. Un problema añadido en Optimización y, en particular, en MINLP, es la búsqueda de soluciones factibles, y unido a éste, el poder asegurar que un problema dado es no factible. Ya hemos mencionado algunas técnicas pensadas para generar soluciones iniciales y ser aplicadas de manera conjunta con otras, como es el caso de la heurística FP. Otra heurística que se utiliza como método de arranque auxiliar en algunos algoritmos MINLP (véase también el punto siguiente) es la *scatter search* (SS; *búsqueda dispersa*) de Glover (1977).

Métodos multiarranque. Los métodos multiarranque consisten, en esencia, en ejecutar varias veces el algoritmo principal de búsqueda, tomando cada vez como solución inicial un punto diferente, con el fin de obtener distintos candidatos a óptimo. Estos métodos son útiles, por ejemplo, cuando se desea mejorar el resultado obtenido al aplicar métodos de MINLP convexa a problemas no convexos, o, en general, siempre que se desee utilizar un algoritmo de búsqueda cuya solución final dependa, en mayor o menor medida, de la solución de partida que se le proporcione.

Se diferencian sobre todo en el modo en como se eligen los puntos iniciales: de manera aleatoria, de manera determinista, etc.

Entre los algoritmos de MINLP que incorporan un método multi-arranque podemos citar OptQuest/NLP, o simplemente OQNLP (véase también el punto siguiente), implementado por [Ugray et al. \(2005\)](#). OQNLP utiliza OptQuest (una versión comercial de la heurística SS arriba mencionada; [Laguna y Martí, 2005, 2002](#)) como método auxiliar para generar un conjunto diverso de soluciones de arranque.

Hibridación de técnicas

Muchos de los enfoques descritos en las subsecciones anteriores se presentan en la práctica implementados de manera híbrida con otras técnicas, con el fin de mejorar el rendimiento final. Es el caso, ya citado, de la técnica de QG (híbrido entre B&B y OA), y también el del trabajo de [Zhu y Kuno \(2003\)](#) (hibridación del B&B con una mejora de la GBD), por ejemplo. Por otro lado, como ya se ha dicho, los tipos de corte que incorpora el método del ECP son utilizados por [Abhishek et al. \(2010a\)](#) para acelerar la convergencia el método de QG. [Bonami et al. \(2008\)](#) diseñan un algoritmo híbrido entre el B&B y la técnica de QG, implementado en el *solver* GAMS/Bonmin (véase la subsección 3.5.2).

Otro ejemplo ya mencionado es el del algoritmo heurístico de [Viswanathan y Grossmann \(1990\)](#), el cual incorpora una función de penalización a la técnica de OA. Asimismo, [Kesavan y Barton \(2000\)](#) proponen combinar su método exacto para MINLP no convexa con una serie de heurísticas auxiliares que consiguen mejorar su rendimiento.

[Cheung et al. \(1997\)](#) combinan un algoritmo genético con una búsqueda tipo *grid*. [Wah y Chen \(2003\)](#) proponen un algoritmo heurístico para MINLP que hibrida *simulated annealing* con algoritmos genéticos. [Xiong y Rao \(2004\)](#) presentan un algoritmo genético para MINLP que utiliza lógica borrosa (*fuzzy logic*).

Por último, el algoritmo heurístico OQNLP ([Ugray et al., 2005](#)) ya

mencionado combina la capacidad de búsqueda global propia de la heurística SS con la habilidad (de búsqueda local) de los métodos de NLP basados en gradiente para mejorar las soluciones de partida. Inicialmente [Ugray et al.](#) lo implementan usando un método clásico de NLP como es el del gradiente reducido generalizado (GRG; [Lasdon et al., 1978](#)), pero aclaran que es posible configurar el método con cualquier algoritmo de NLP, incluso de los que no usan derivadas.

Esta es sólo una muestra de la gran variedad de posibles hibridaciones.

Información adicional

Explicar más en profundidad el funcionamiento y los detalles de los algoritmos que acabamos de presentar queda fuera de los objetivos de esta memoria. [Belotti et al. \(2013\)](#) proporcionan una excelente y exhaustiva revisión del desarrollo de algoritmos para MINLP. Asimismo, para profundizar sobre estos y otros aspectos generales de la MINLP pueden consultarse, entre otros, los trabajos de [Leyffer \(1993\)](#), [Floudas \(1995\)](#), [Grossmann \(2002\)](#), [Bussieck y Pruessner \(2003\)](#), [Biegler y Grossmann \(2004\)](#), [Caballero y Grossmann \(2007\)](#), [Bussieck y Vigerske \(2011\)](#), [Bonami et al. \(2012\)](#), [Burer y Letchford \(2012\)](#) y [Lee y Leyffer \(2012\)](#).

1.5. Objetivos y estructura de la tesis

En esta tesis doctoral abordamos el diseño óptimo de planes de precontrol, desde su modelización hasta su puesta en práctica. Tomaremos como punto de partida la variante generalizada de dicha técnica propuesta por [San Matías et al. \(2004\)](#), en la cual, como acabamos de destacar, se *libera* el valor de ciertos parámetros que estaban fijados en la versión clásica, proporcionando al precontrol una mayor flexibilidad sin restarle sencillez. Mediante el presente trabajo pretendemos dar una respuesta desde la Investigación Operativa a la pregunta de *cuál es la manera óptima de configurar los parámetros que rigen el comportamiento del precontrol*, teniendo como meta no sólo contestar de manera teórica sino también proporcionar

una herramienta para poder obtener, en la práctica, planes de precontrol óptimos a partir de unas premisas o condiciones iniciales dadas.

Por tanto, el **primer objetivo** de la tesis es *plantear formalmente la determinación de los parámetros del precontrol como problema de optimización*. Así, en el **capítulo 2** describimos con detalle la técnica del precontrol y realizamos una revisión completa de las diferentes variantes de la misma que han surgido en la literatura, y en particular del precontrol generalizado previamente aludido. De esta manera, estaremos en condiciones de abordar en el **capítulo 3** la formalización del problema del precontrol óptimo mediante la elaboración de un modelo de programación matemática. En concreto, veremos que el modelo obtenido pertenece al ámbito de la programación no lineal entera mixta. En el mismo capítulo examinamos los algoritmos existentes para resolver dicha clase de modelos, y los aplicamos sobre una batería de ejemplos para evaluar su comportamiento sobre nuestro problema.

Nuestro **segundo objetivo** es *diseñar e implementar algoritmos específicos para el problema del precontrol óptimo*, como alternativa a la utilización de *software* de optimización de propósito general. Por ello, y con el fin de tener un mayor conocimiento sobre el problema concreto que se desea resolver, en el **capítulo 4** de la tesis estudiamos de manera analítica el comportamiento de las funciones involucradas en el modelo del precontrol óptimo que hemos definido. A continuación, en el **capítulo 5** desarrollamos nuestra propuesta, consistente en un nuevo algoritmo exacto para la obtención de planes de precontrol óptimos, basado en una estrategia enumerativa. El algoritmo ha sido implementado y posteriormente evaluado sobre la misma batería de ejemplos previamente utilizada.

En el planteamiento del precontrol como problema de optimización y en el algoritmo que hemos diseñado se mantienen las hipótesis mínimas *heredadas* de los trabajos previos a nuestra propuesta, como son, por un lado, la normalidad de la característica de calidad de interés, y la simetría de los límites de precontrol y de especificaciones respecto del valor nominal, por otro. De hecho, dichas hipótesis son habituales en el ámbito

del control de la calidad y, más concretamente, en la monitorización de la estabilidad de procesos. Con todo, consideramos que la idea del precontrol óptimo podría adaptarse a otros supuestos y contextos.

Por ello, en el **capítulo 6** de la tesis nos planteamos *estudiar de manera preliminar la extensión del precontrol óptimo a otras distribuciones de probabilidad y explorar su aplicación a otros ámbitos más allá del control de la calidad*, como por ejemplo la fiabilidad y el mantenimiento de sistemas. Más concretamente, extendemos el modelo del precontrol óptimo a un caso más general en el que la distribución de probabilidad subyacente no necesariamente sea normal ni los límites de control tengan por qué ser simétricos, y mostramos la utilidad de este nuevo planteamiento mediante su aplicación a la monitorización de fallos de un proceso. Se trata de una primera aproximación a este nuevo enfoque, que se completa con un resumen de otras aportaciones propias en relación al precontrol. En dicho capítulo detallamos, además, las contribuciones en congresos y publicaciones científicas relacionadas con esta tesis.

Para finalizar, en el **capítulo 7** recogemos las conclusiones que se derivan de todo el trabajo realizado y proponemos líneas futuras de investigación.

Precontrol. Origen y estado actual

En este capítulo presentamos la técnica del precontrol, como herramienta para el control de la calidad. Describiremos con detalle el funcionamiento de la versión clásica de esta técnica, y señalaremos sus puntos fuertes y sus inconvenientes.

Posteriormente, se revisarán las alternativas a dicha técnica propuestas en la literatura con el fin de mejorar su comportamiento, deteniéndonos en la versión propuesta por [San Matías et al. \(2004\)](#). Analizaremos las mejoras que supone dicha versión sobre el precontrol clásico y las ilustraremos mediante un ejemplo práctico.

De esta manera, introducimos los elementos en los que nos basaremos para elaborar y desarrollar nuestra propuesta en los capítulos posteriores.

2.1. La técnica del precontrol

2.1.1. Definición y origen del precontrol

El precontrol es una técnica de calidad que se usa para evaluar la capacidad o aptitud de un proceso de producción para generar unidades que se adecuen a unas especificaciones antes fijadas, en ocasiones de manera previa al inicio de un control estándar del proceso.

Como ya se ha mencionado en la introducción, el precontrol fue

desarrollado en 1952 por Frank [Satterthwaite \(1954\)](#) y un equipo formado, además de por él mismo, por Charles W. Carter, Warren R. Purcell y Dorian Shainin, como alternativa a los gráficos de control \bar{X} - R de [Shewhart \(1931\)](#), en contextos en que el uso de éstos no es adecuado o no es factible: por ejemplo, procesos con ciclos de ejecución cortos. A diferencia de los gráficos de control, se trata de un método cuyo objeto es supervisar un proceso teniendo en cuenta directamente las especificaciones del producto. [Shainin y Shainin \(1989\)](#) recuperan y publican la versión original del precontrol, e informan del éxito conseguido en la reducción de defectos por su amplio uso en muchas empresas. También es conocido como *stoplight control* ([Heaphy y Gruska, 1984](#); [Salvia, 1988](#)) y *target area control* ([Brown Jr, 1966](#)).

2.1.2. Precontrol clásico

El objetivo del precontrol es detectar cuándo una proporción inaceptable de piezas se sitúa fuera de las especificaciones. A continuación pasamos a describir con detalle el procedimiento para llevar a cabo el precontrol de un proceso.

Supongamos que interesa controlar una característica de calidad X unidimensional y continua de media $E[X] = \mu$, cuyo valor nominal o valor objetivo es μ_0 , y sean LSL y USL sus límites de especificación inferior y superior, respectivamente. Asumiremos que μ_0 se encuentra en el centro del rango de tolerancias, dado que es así en muchos procesos. Así:

$$\mu_0 = \frac{\text{LSL} + \text{USL}}{2}. \quad (2.1)$$

Método de clasificación

En primer lugar, se definen dos límites, G_L y G_U , en el punto medio entre el valor nominal y los límites de tolerancia. Es decir:

$$G_L = \frac{\text{LSL} + \mu_0}{2} \quad \text{y} \quad (2.2)$$

$$G_U = \frac{\mu_0 + USL}{2}. \quad (2.3)$$

Estos valores reciben el nombre de *líneas de precontrol* o *límites de precontrol*, y dividen el rango de tolerancias en varias zonas: una zona central entre G_L y G_U , que suele recibir el nombre de *zona verde*, y dos áreas laterales, que conforman la *zona de precaución* o *zona amarilla*. Por último, se define la *zona roja* como el conjunto de valores fuera de especificaciones (véase la figura 2.1).

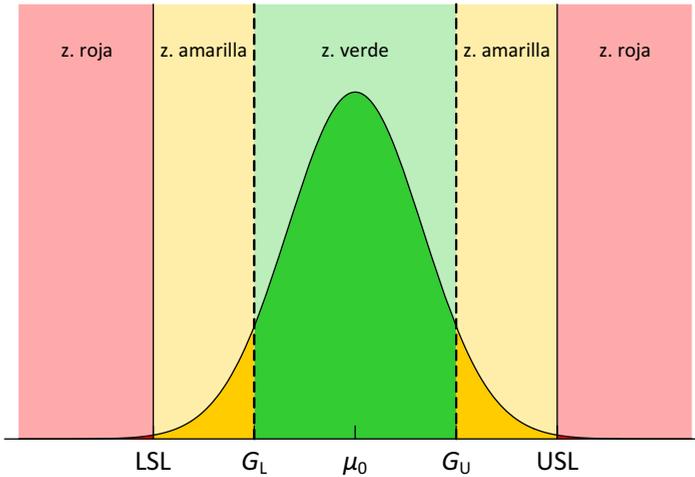


Figura 2.1: Método de clasificación del precontrol clásico, para una característica de calidad con distribución normal, centrada y con capacidad potencial $C_p = 1$.

De manera equivalente, al muestrear una pieza, diremos que es una pieza *verde* si su valor observado de X pertenece a la zona central, *amarilla* si cae dentro de la zona de precaución, o *roja* si está fuera de la zona de especificaciones.

Algoritmo del precontrol clásico

Se suele denominar *precontrol clásico* a la versión original del precontrol, tal y como es descrito en [Shainin y Shainin \(1989\)](#).

El precontrol clásico consta de dos etapas: la *fase de cualificación*, en que se valida la puesta a punto del proceso, y otra a la que nosotros llamaremos *fase de mantenimiento*, en la cual se revisa de manera periódica si se siguen produciendo piezas dentro de especificaciones.

Durante la fase de cualificación, se muestrean piezas sucesivamente. Esta fase finaliza cuando se obtienen 5 piezas verdes consecutivas, en cuyo caso se dice que el proceso está *cualificado*. Si son muestreadas 2 piezas amarillas consecutivas o 1 roja, el proceso se considera *no cualificado*; en ese caso, se detiene el proceso, se revisa y ajusta, y vuelve a iniciarse la fase de cualificación.

La fase de mantenimiento tiene lugar durante la ejecución normal del proceso, una vez ha sido cualificado en la fase anterior. Periódicamente se muestrea una pieza. Si ésta es verde, se considera que el proceso no necesita ningún ajuste y puede continuar al menos hasta la próxima vez que se muestree otra pieza. Si la pieza muestreada ha sido roja, el proceso necesita ser detenido para su ajuste, y, posteriormente, volver a la fase de cualificación. Si la pieza muestreada ha sido amarilla, se muestrea una segunda pieza. Si esta segunda pieza no es verde, el proceso se considera también desajustado, siendo necesario detenerlo, ajustarlo y volver a la fase de cualificación.

Traver (1985), entre otros autores, puntualiza que si, en la fase de mantenimiento, dos piezas amarillas se obtienen en la misma subzona (es decir, las dos por arriba del valor nominal o las dos por debajo), hay que sospechar que la media del proceso se haya desajustado, mientras que si cada pieza amarilla se observa en una subzona diferente, puede tomarse como un indicio de que la variabilidad del proceso ha aumentado.

La frecuencia de muestreo de la fase de mantenimiento se puede ir adecuando a través de la experiencia.

Los algoritmos 2.1 y 2.2 muestran en pseudocódigo el procedimiento correspondiente al precontrol clásico; también se recoge de forma esquemática en la figura 2.2.

Algoritmo 2.1: Precontrol clásico – Fase de cualificación

Paso 1 // Inicialización $m = 0$ // Número de verdes consecutivas $h = 0$ // Número de amarillas consecutivas**Paso 2** // Muestreo intensivo

Extraer una pieza

si la pieza es verde **entonces** $m = m + 1$ $h = 0$ **ir a Paso 3****si** la pieza es amarilla **entonces** $m = 0$ $h = h + 1$ **ir a Paso 3****si** la pieza es roja **entonces** $m = 0$ $h = 0$ **DETENER; mostrar** "Proceso NO cualificado"Ajustar el proceso; **ir** al inicio de **Fase de Cualificación****Paso 3****si** $m < 5$ y $h < 2$ **entonces**

// Seguir muestreando

ir a Paso 2**en otro caso****ir a Paso 4****Paso 4****si** $m = 5$ **entonces** // 5 verdes consecutivas**DETENER; mostrar** "Proceso cualificado"**ir** al inicio de **Fase de Mantenimiento****en otro caso** // ($h = 2$) 2 amarillas consecutivas**DETENER; mostrar** "Proceso NO cualificado"Ajustar el proceso; **ir** al inicio de **Fase de Cualificación**

Algoritmo 2.2: Precontrol clásico – Fase de Mantenimiento

Paso 1 // Primera extracción

Extraer una pieza

si la pieza es verde **entonces**

DETENER; mostrar *"Proceso ajustado"*

Esperar hasta la próxima medición programada; **ir** al inicio de **Fase de Mantenimiento**

si la pieza es roja **entonces**

DETENER; mostrar *"Proceso NO ajustado"*

Ajustar el proceso; **ir** al inicio de **Fase de Cualificación**

si la pieza es amarilla **entonces**

ir a Paso 2

Paso 2 // Segunda extracción

Extraer una segunda pieza

si la segunda pieza no es verde **entonces**

DETENER; mostrar *"Proceso NO ajustado"*

Ajustar el proceso; **ir** al inicio de **Fase de Cualificación**

en otro caso

DETENER; mostrar *"Proceso ajustado"*

Esperar hasta la próxima medición programada; **ir** al inicio de **Fase de Mantenimiento**

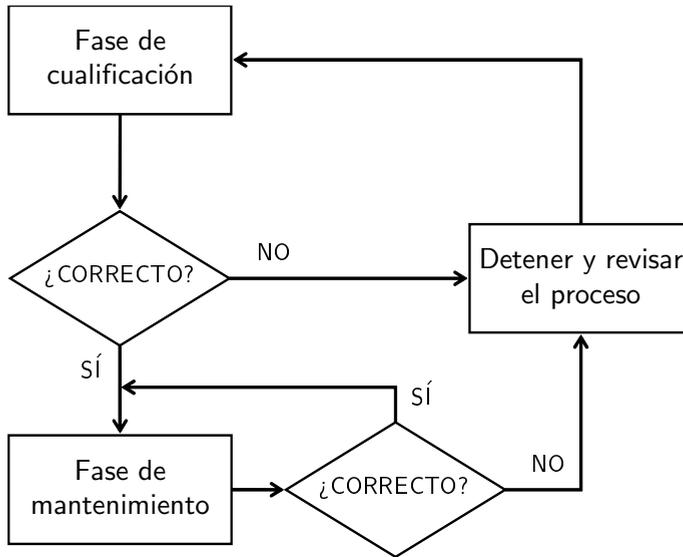


Figura 2.2: Esquema de aplicación del precontrol clásico. Este esquema es respetado por todas las variantes de la técnica que se exponen en el presente capítulo.

2.1.3. Aplicación del precontrol

De acuerdo con [Shainin y Shainin \(1989\)](#) y [Bhothe \(1988\)](#), el precontrol es utilizado por un gran número de compañías, orientado al control basado en las tolerancias, y visto siempre como un método para series de producción cortas y rápidas. La forma más frecuente de implementar el precontrol ([Olorunniwo y Knight, 1995](#)) es mediante el uso de dispositivos del estilo de *calibres pasa/no pasa* (*go-no-go gauge*, en inglés) o similares, normalmente ya marcados con los colores verde, amarillo y rojo (véase la [figura 2.3](#)), de modo que se facilite al operario o usuario final la tarea de registrar el resultado de cada observación¹.

¹De hecho, el precontrol puede ser visto como un caso particular de *calibración de límites estrechos* (*compressed-limit gauging*, *narrow-limit gauging* o *narrow gauge charts*, en inglés; [Dudding y Jennett, 1944](#), [Ott y Mundel, 1954](#), [Trietsch, 1999](#), [Montgomery, 2012](#)), una técnica en la que se utilizan calibres con especificaciones muy ajustadas (las que delimitarían la zona verde, en el caso del precontrol), generalmente con el fin de asegurar

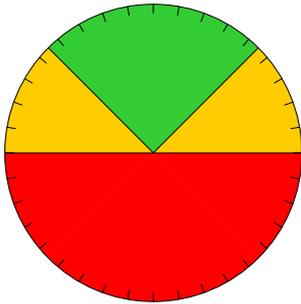


Figura 2.3: Carátula para la esfera de un reloj comparador analógico con las zonas verde, amarilla y roja delimitadas. Elaboración propia a partir de una imagen de [Shainin y Shainin \(1989\)](#).

El modo típico de aplicar el precontrol no requiere, por tanto, almacenaje de datos ni cálculos adicionales por parte del usuario final (véase más adelante la subsección 2.1.4), ni la realización de ningún gráfico *a escala* (como sí sucede en los gráficos de control). En ocasiones resulta conveniente y visualmente más claro recoger las observaciones en un gráfico simplificado, con tres zonas diferenciadas, verde, amarilla y roja (o cinco, si se quiere distinguir entre las piezas amarillas o rojas producidas por arriba y por debajo del valor nominal), pero no es necesario graficar ni registrar el valor

numérico exacto de cada observación ([Olorunniwo y Knight, 1995](#)). Aun así, [Sugg y Miller \(1992\)](#) sí recomiendan graficar los datos del precontrol, con el fin de poder detectar posibles «patrones sutiles». La figura 2.4 muestra diversas maneras de registrar los resultados del procedimiento de precontrol.

Como curiosidad, un caso histórico de la aplicación del precontrol y de los resultados obtenidos puede encontrarse en [Brown Jr \(1966\)](#). Además del conocido uso del precontrol en la industria y de las propuestas de mejora de la técnica clásica que se detallarán en las secciones siguientes, aparecen en la literatura no pocas referencias a usos del precontrol en los campos más diversos. Destacan las aplicaciones en el ámbito de los estudios clínicos ([Minne et al., 2012a,b](#)) y al control presupuestario en fábricas y empresas ([Krehbiel et al., 2007, 2013](#)). [Vermani \(2000\)](#) y [Torbeck \(2005\)](#) presentan adaptaciones del precontrol a sendos casos en las industrias aeronáutica militar y farmacéutica, respectivamente; ambas propuestas son desarrolladas en la sección 2.3.

una cierta probabilidad de rechazo ante una partida con una determinada desviación inaceptable, en el contexto de muestreos de aceptación.

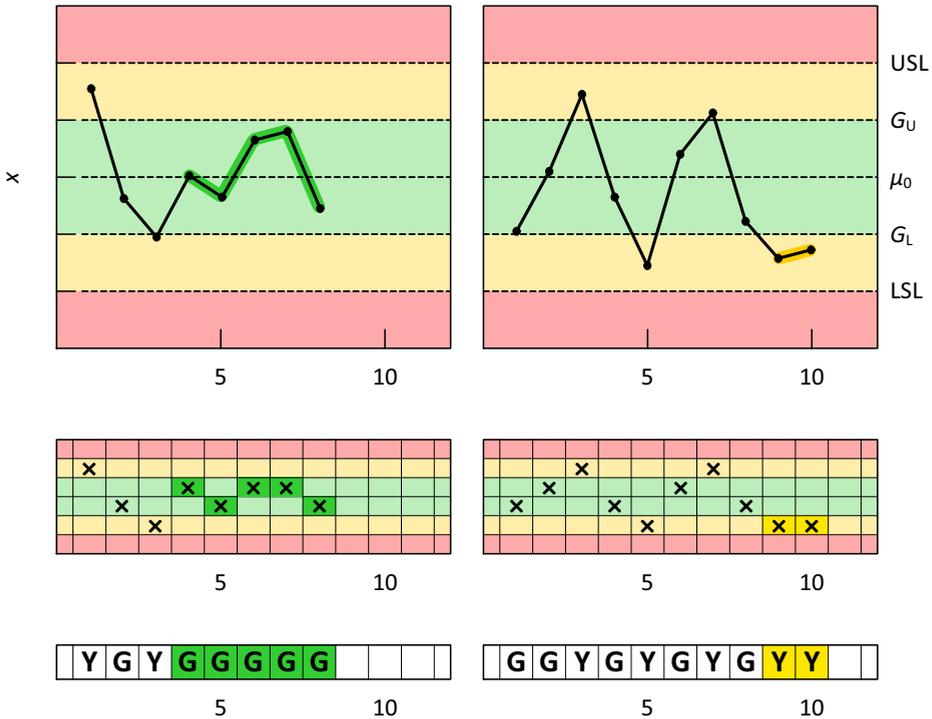


Figura 2.4: Gráficos de precontrol para dos ejecuciones de la técnica: a la izquierda, un proceso que es cualificado tras observar 5 piezas verdes consecutivas; a la derecha, uno que no es cualificado por obtenerse 2 amarillas consecutivas. Se muestran el gráfico a escala (arriba), el gráfico simplificado con cinco zonas (medio; el más habitual), y una versión compacta (abajo). Elaboración propia.

2.1.4. Ventajas y puntos débiles del precontrol clásico

Precontrol frente a los gráficos de control

Según varios autores (Traver, 1985; Martin, 1993; Trietsch, 1999), el precontrol salva algunos de los inconvenientes que muestran los gráficos de control \bar{X} -R cuando son aplicados en ciertos contextos.

En primer lugar, como se acaba de destacar, el precontrol no precisa para ser llevado a cabo la recopilación y almacenamiento de datos ni su posterior tratamiento. Esto supone una ventaja frente a los gráficos de

control, sobre todo en situaciones en que el ciclo de producción es corto y no da tiempo a calcular los límites de control o a dibujar los gráficos.

Además, por otro lado, el precontrol trabaja directamente con las especificaciones del producto, mientras que los gráficos de control tienen en cuenta los límites de control calculados. Los gráficos de control pueden generar señales de falta de control, ligadas a la falta de estabilidad, incluso cuando la totalidad de la producción se encuentra dentro de las especificaciones, sobre todo en procesos con una alta capacidad (es decir, aquellos en que la variabilidad del proceso es pequeña en comparación con las tolerancias). En cambio, el precontrol no detecta este tipo de situaciones. Ello puede suponer una ventaja en aquellos casos en que la estabilidad del proceso no es importante y en que baste con asegurar el cumplimiento de las especificaciones.

Por último, el precontrol puede ser utilizado incluso si la característica de calidad a controlar se trata de un atributo (es decir, una característica discreta) o una cualidad visual que no puede ser medida numéricamente, casos en los que, obviamente, no es adecuado usar un gráfico de control \bar{X} - R (sino uno de control para atributos). Supone, pues, la ventaja de poder utilizar un único procedimiento tanto para variables (características numéricas continuas) como para atributos.

Cabe decir que el nombre de *precontrol* no significa que este procedimiento deba aplicarse necesariamente antes de que el proceso dé comienzo, aunque éste sea uno de sus usos, sino que hace referencia al hecho, ya mencionado, de que permite detectar problemas en el proceso antes de que un gran número de unidades defectuosas sean generadas (Traver, 1985), sobre todo en ejecuciones rápidas, situaciones en las que podría decirse que no es práctico el uso de gráficos de control (Trietsch, 1999).

Martin (1993) apoya esta tesis y, ante el enfrentamiento entre los partidarios de una u otra técnica, concluye que el precontrol no puede sustituir a los gráficos de control —aunque apunta que éstos, a veces, son sobreutilizados—, sino que, sencillamente, se trata de una herramienta distinta, con su propio ámbito de aplicación.

En conclusión, podemos decir que, si bien el precontrol surgió como alternativa a los gráficos de control en ciertos contextos o situaciones en que la aplicación de éstos no es del todo posible y/o adecuada, ambos métodos persiguen objetivos diferentes. El propósito del precontrol es detectar cuándo la proporción de piezas no conformes es inaceptablemente elevada, mientras que los gráficos de control están diseñados para revelar desviaciones del proceso. En este sentido, [Trietsch \(1999\)](#) llega a afirmar que el control estadístico no incumbe al precontrol, ya que éste tiene como objeto evaluar la *capacidad* del proceso, y no su *estabilidad*, como los gráficos de control.

Puede encontrarse una comparación formal entre el precontrol y los gráficos de control en [Ledolter y Swersey \(1997\)](#).

A modo de resumen, podemos destacar que el precontrol supone una herramienta para el control de un proceso fácil de utilizar, no requiere muestrear una gran cantidad de datos y está directamente relacionado con el cumplimiento de las especificaciones del producto.

Precontrol frente a los gráficos de aceptación

Los gráficos de control de aceptación fueron introducidos por [Freund \(1957\)](#) y posteriormente revisados por [Woods \(1976\)](#). Algunos autores (véase [Holmes y Mergen, 2000](#), entre otros) sugieren nombrar esta técnica simplemente como *gráficos de aceptación*, dado que su objetivo no es evaluar si un proceso se encuentra bajo control, sino si produce una proporción inaceptable de piezas fuera de especificaciones (véase la subsección [1.2.2](#)). Por tanto, este tipo de gráficos *sí* coincide en su propósito con el precontrol.

En esta técnica, al igual que en los gráficos de control de Shewhart, se representan gráficamente las medias muestrales a lo largo del tiempo, pero, a diferencia de lo que sucede en el gráfico \bar{X} tradicional, los límites de control no están basados en la variabilidad del proceso ($\mu \pm 3\sigma$, habitualmente), sino en las especificaciones o tolerancias del producto. Se definen unos límites tales que se asegure una cierta probabilidad $1 - \alpha$ de aceptar un proceso con una proporción aceptable dada de piezas

disconformes y una cierta probabilidad $1 - \beta$ de rechazar un proceso con una proporción inaceptable dada de piezas disconformes (de manera similar a como se procede, por ejemplo, al diseñar un plan de aceptación muestral). Dichos límites no son llamados *límites de control*, pues este término se reserva para los gráficos propiamente de control, sino que reciben el nombre de *límites de aceptación*.

Estos gráficos resultan adecuados cuando el interés no está en detectar desviaciones del proceso sino proporciones inaceptables de piezas no conformes, y especialmente en procesos *muy capaces* (es decir, con capacidad potencial C_p mayor que 1,3333 o 1,6667), en los que una señal de falta de control puede no suponer un aumento inaceptable de la proporción de piezas no conformes, no siendo la estabilidad un requisito.

Steiner (1997) compara en profundidad la técnica de los gráficos de aceptación con la fase de mantenimiento del precontrol. En primer lugar, muestra a través de las respectivas curvas características que los gráficos de aceptación (con tamaño muestral $n = 2$) presentan mayor potencia que la fase de mantenimiento del precontrol para detectar desviaciones de la media del proceso.

Además, el precontrol sólo considera las especificaciones para definir sus límites, de modo que, según señala Steiner, procesos con una desviación de la media relativamente elevada y una variabilidad muy pequeña (es decir, una capacidad potencial C_p muy alta) van a provocar con bastante probabilidad señales de alerta (como resultado de muestrear dos piezas amarillas consecutivas en la fase de cualificación o en la de mantenimiento) sin que necesariamente dicho proceso esté generando una proporción inaceptable de piezas no conformes. En cambio, por el modo en como están diseñados, los gráficos de aceptación tienen en cuenta la variabilidad real del proceso σ o una estimación de la misma y adaptan sus límites en consecuencia.

A favor de los gráficos de aceptación también está el hecho de que requieren un tamaño muestral constante. En cambio, presentan como desventajas frente al precontrol (Steiner, 1997) la necesidad de conocer una estimación de σ , de suponer que ésta se mantiene constante, y de

realizar mediciones exactas en lugar de una simple clasificación en tres niveles (verde, amarillo y rojo).

Por otro lado, **Steiner** recoge la afirmación de **Traver (1985)**, basada en la experiencia, según la cual el precontrol clásico debería ser usado sólo cuando la variabilidad natural del proceso 6σ sea igual o inferior a $88\%(USL - LSL)$ (lo que equivale a decir que la capacidad potencial C_p del proceso debería ser igual o superior a 1,125), para evitar la aparición excesiva de piezas amarillas y rojas y los consiguientes ajustes innecesarios.

De todo ello, **Steiner** concluye que el precontrol es bastante competitivo frente a los gráficos de aceptación, debido a su sencillez, excepto en los casos en que la variabilidad del proceso es muy pequeña en comparación con el rango de tolerancias (concretamente, para procesos con $C_p > 1,667$, sin que **Steiner** justifique dicha cifra en concreto), en cuyo caso el precontrol puede generar, como se ha explicado, una señal de alarma en situaciones en que prácticamente todas las piezas están dentro de especificaciones, y salvo también cuando la variabilidad del proceso es elevada en relación con las tolerancias (en concreto, para $C_p < 1,125$, como ya se ha mencionado).

Estas conclusiones son válidas para el precontrol clásico y algunas variaciones de éste que **Steiner** analiza, y que describiremos a continuación. Más adelante presentamos una generalización del precontrol propuesta por **San Matías et al. (2004)** que, entre otras mejoras, permite reducir la tasa de falsa alarma.

En cualquier caso, y sin tener en cuenta las mejoras posteriores, el precontrol clásico presenta frente a los gráficos de aceptación la ventaja de la sencillez y la rapidez en el procedimiento y la no exigencia de un elevado tamaño muestral.

2.2. Alternativas al precontrol clásico

Desde la aparición del precontrol como propuesta alternativa a los gráficos de control, se han sucedido diferentes versiones del mismo,

las cuales pasamos a discutir en esta sección. Un completo análisis de algunas alternativas al precontrol clásico puede encontrarse en [Steiner \(1997\)](#). Las versiones del precontrol que analizaremos y compararemos son las siguientes:

- Precontrol en dos etapas ([Salvia, 1988](#)).
- Precontrol modificado ([Gruska y Heaphy, 1991](#)).
- Precontrol de diez unidades ([Steiner, 1997](#)).
- Precontrol del cambio de media ([Steiner, 1997](#)).
- Precontrol simplificado ([Steiner, 1997](#)).
- *Pairs* ([Chamberlayne, 1997](#)).
- Precontrol modificado de [Družovec et al. \(1998\)](#).
- *Set-Up Process Algorithm* ([Cox et al., 2013b](#)).

El *precontrol en dos etapas* ([Salvia, 1988](#)) y el *precontrol modificado* ([Gruska y Heaphy, 1991](#)) son las dos primeras propuestas surgidas con el objeto de mejorar la técnica original y, en el caso del precontrol modificado, de incorporar a ésta la filosofía de los gráficos de control. Junto con el precontrol clásico, pueden agruparse bajo la etiqueta de *precontrol tradicional*.

El *precontrol de diez unidades*, el *precontrol del cambio de media* y el *precontrol simplificado* son tres versiones sugeridas por [Steiner \(1997\)](#), que incorporan pequeñas modificaciones sobre las técnicas tradicionales para intentar mejorar su comportamiento, sin perder de vista la simplicidad del procedimiento.

Describiremos también otras modificaciones del precontrol, debidas a [Chamberlayne \(1997\)](#) y [Družovec et al. \(1998\)](#). Esta última, al igual que la propuesta de [Gruska y Heaphy \(1991\)](#), recibe el nombre de *precontrol modificado* (en inglés, *modified pre-control*; MPC).

Por último, se detallará una propuesta reciente de [Cox et al. \(2013b\)](#). Tanto esta como las de [Chamberlayne](#) y [Družovec et al.](#) plantean mejoras en la fase de cualificación del precontrol, mientras que las anteriores se centran principalmente en la fase de mantenimiento.

Puede afirmarse que un esquema de precontrol queda totalmente definido cuando se determinan los tres aspectos siguientes:

Método de clasificación: Procedimiento utilizado para clasificar las piezas que se muestrean.

Fase de cualificación: Diseño de la fase de cualificación.

Criterio de decisión: Criterio utilizado en la fase de mantenimiento para decidir si el proceso necesita o no ser revisado y ajustado.

Las diferentes versiones del precontrol que a continuación vamos a describir surgen a partir de variaciones sobre el precontrol clásico en todos o algunos de estos tres aspectos.

2.2.1. Precontrol en dos etapas

El precontrol en dos etapas ([Salvia, 1988](#)) comparte con el precontrol clásico el método de clasificación de las unidades en rojas, amarillas y verdes, así como el diseño de la fase de cualificación.

El criterio de decisión usado en la fase de mantenimiento es, sin embargo, un poco más elaborado en el precontrol en dos etapas, con la finalidad de conseguir una decisión más acertada en casos dudosos. Se muestrean 2 piezas consecutivas. El proceso se detiene si alguna de ellas es roja, y se considera ajustado si las dos son verdes². En otro caso (una o

²Aunque esta es la forma habitual de describir el precontrol en dos etapas, obviamente, si la primera pieza extraída en esta primera etapa resultase ser roja, no haría falta extraer la segunda pieza para dar por finalizado el procedimiento. Este detalle es relevante en el caso de que, por ejemplo, se esté interesado en estimar el tamaño muestral medio necesario para llevar a cabo la fase de mantenimiento de esta variante del precontrol.

Algunas de las variantes que se detallarán a continuación también se describen en los mismos términos (muestrear 2, 3, 5 o 6 piezas consecutivas), y en todas ellas hay que entender que si algún criterio de parada se cumple antes de finalizar el muestreo, no hace falta continuar con él.

dos amarillas y ninguna roja), se muestrean hasta un máximo de 3 piezas más. Si aparece una pieza roja o se llega en algún momento a acumular 3 amarillas, el proceso se detiene para ser revisado y ajustado. Si, por el contrario, el número de piezas verdes acumuladas llega a ser 3, el proceso se considera ajustado. El algoritmo 2.3 muestra la fase de mantenimiento que acabamos de describir.

Steiner (1997) extiende al precontrol en dos etapas la norma de Traver (1985), ya mencionada en la subsección 2.1.4, según la cual el precontrol clásico debería ser usado sólo cuando la capacidad potencial C_p del proceso sea igual o superior a 1,125, para evitar ajustes innecesarios.

Steiner compara la fase de mantenimiento del precontrol en dos etapas con los gráficos de aceptación (véase la subsección 1.2.2). En términos de potencia, el precontrol en dos etapas es superior a un gráfico de aceptación con tamaño muestral $n = 2$, y no mucho peor que uno con $n = 3$, de modo que Steiner termina considerando el precontrol en dos etapas como una buena alternativa a los gráficos de aceptación cuando no es posible estimar la media y la desviación del proceso (paso necesario para poder definir los límites de dichos gráficos) o es muy costoso.

En cualquier caso, al igual que con el precontrol clásico (véase la subsección 2.1.4), Steiner recomienda restringir el uso del precontrol en dos etapas a situaciones en que $1,125 \leq C_p \leq 1,667$. A pesar de ser más complejo y requerir, en media, un mayor tamaño muestral que el precontrol clásico en la fase de mantenimiento, su ventaja frente a éste es que recaba más información del proceso y, por tanto, es capaz de reducir la proporción de decisiones erróneas.

2.2.2. Precontrol modificado

El precontrol modificado (Gruska y Heaphy, 1991) fue propuesto con la idea de mantener la simplicidad del esquema clásico, recuperando al mismo tiempo un enfoque de gráficos de control. Así, comparte con los dos anteriores la fase de cualificación, y mantiene el mismo criterio de decisión que el precontrol en dos etapas.

Sin embargo, para definir las zonas roja, amarilla y verde el precontrol

Algoritmo 2.3: Precontrol en dos etapas – Fase de mantenimiento

Paso 1 // Inicialización $m = 0$ // número acumulado de verdes en el total de piezas muestreadas $h = 0$ // número acumulado de amarillas en el total de piezas muestreadas $r = 0$ // número acumulado de rojas en el total de piezas muestreadas**Paso 2**

Extraer 2 piezas consecutivas

si $m = 2$ **entonces** // las dos piezas son verdes**DETENER; mostrar** "*Proceso ajustado*"Esperar hasta la próxima medición programada; **ir** al inicio de **Fase de Mantenimiento****si** $r \geq 1$ **entonces** // alguna de las dos piezas es roja**DETENER; mostrar** "*Proceso NO ajustado*"Ajustar proceso; **ir** al inicio de **Fase de Cualificación****en otro caso** // 2 amarillas, o 1 verde y 1 amarilla**ir a Paso 3****Paso 3**

Extraer 1 pieza

Actualizar m , h y r // actualizamos los acumulados, sobre el total de piezas muestreadas hasta el momento**si** $m = 3$ **entonces** // hemos llegado a acumular 3 verdes sin observar ninguna roja**DETENER; mostrar** "*Proceso ajustado*"Esperar hasta la próxima medición programada; **ir** al inicio de **Fase de Mantenimiento****si** $h = 3$ o $r = 1$ **entonces** // hemos observado ya un total de 3 amarillas, o bien acabamos de muestrear una roja**DETENER; mostrar** "*Proceso NO ajustado*"Ajustar proceso; **ir** al inicio de **Fase de Cualificación****en otro caso****ir a Paso 3**

modificado hace uso de los límites de control (de manera similar a como se definen en los gráficos \bar{X} - R), en lugar de las especificaciones del producto. Concretamente, si consideramos que el proceso bajo control presenta para la variable de interés una media μ_0 y una desviación típica σ_0 , la zona verde será la formada por los valores que difieran de μ_0 en menos de $1,5\sigma_0$, es decir:

$$G_L = \mu_0 - 1,5\sigma_0, \quad (2.4)$$

$$G_U = \mu_0 + 1,5\sigma_0. \quad (2.5)$$

La zona amarilla se define como aquellos valores que distan de μ_0 entre 1,5 y 3 veces la desviación típica σ_0 . Por último, la zona roja, estará formada por los valores cuya distancia a μ_0 sea superior a $3\sigma_0$.

Hacemos notar que, al igual que en los otros dos esquemas tradicionales ya descritos, la anchura de la zona verde sigue siendo la mitad de la anchura total del intervalo formado por la unión de las zonas verde y amarilla.

En procesos en que la capacidad bajo control, $6\sigma_0$, coincida con la anchura de la zona de especificaciones (es decir, cuando la capacidad potencial del proceso C_p cuando está bajo control es 1), el precontrol modificado es equivalente al esquema en dos etapas.

Como se ha dicho, la ventaja de esta variante es que recupera el enfoque de los gráficos de control. Sin embargo, al mismo tiempo este hecho hace que se pierda una de las ventajas fundamentales del precontrol, como es su adecuación directa a las especificaciones del producto, que aquí son obviadas. Se trata, por tanto, de una variante del precontrol que se orienta al control de la estabilidad, y no al del cumplimiento de las especificaciones.

Además, al igual que sucede con el gráfico de control \bar{X} tradicional, esta variante precisa para su aplicación de una estimación previa de los parámetros del proceso bajo control, μ_0 y σ_0 . Sin embargo, presenta una mayor tasa de falsa alarma que éste, por lo que en general no se recomienda su uso (Steiner, 1997).

2.2.3. Precontrol de las diez unidades

El precontrol de las diez unidades es la primera de las tres alternativas al precontrol tradicional propuestas por Steiner (1997). Trata de mejorar el comportamiento de la técnica mediante una cierta generalización del enfoque secuencial de la fase de mantenimiento introducido en el precontrol en dos etapas, sin perder de vista la sencillez de todo el procedimiento.

Así, el precontrol de las diez unidades conserva el mismo método de clasificación y el mismo diseño para la fase de cualificación que el precontrol en dos etapas y que el precontrol clásico. En la fase de mantenimiento, sin embargo, se muestrean unidades sucesivamente, de tal manera que el proceso deberá ser revisado si en cualquier momento se obtiene una pieza roja, o bien el número de piezas amarillas extraídas llega a ser de 5, o bien es al menos de 3 y supera en al menos 2 al número de piezas verdes extraídas. El proceso se considera ajustado si en cualquier momento el número de piezas verdes extraídas supera en al menos 2 al de amarillas (véase el algoritmo 2.4).

Este esquema hace que como máximo se muestreen 10 unidades en la fase de mantenimiento. Sin embargo, en general el número esperado de piezas a muestrear no es muy superior al del esquema en dos etapas, mientras que sí presenta una menor tasa de falsa alarma y una mayor potencia para detectar desviaciones importantes de la media del proceso.

2.2.4. Precontrol del cambio de media

Como sabemos, el precontrol tradicional clasifica las piezas muestreadas en verdes, amarillas y rojas. En cambio, el precontrol del cambio de media (Steiner, 1997) distingue entre las zonas amarillas inferior y superior, con lo que considera cuatro zonas en lugar de tres (véase la figura 2.5). El objetivo de esta modificación es poder detectar mejor desviaciones de la media real del proceso, que pueden pasar inadvertidas en el precontrol tradicional.

Es en la fase de mantenimiento de esta variante del precontrol donde

Algoritmo 2.4: Precontrol de las diez unidades – Fase de mantenimiento

Paso 1 // Inicialización

$m = 0$ // número acumulado de verdes

$h = 0$ // número acumulado de amarillas

$r = 0$ // número acumulado de rojas

Paso 2

Extraer 1 pieza

Actualizar m , h y r // actualizamos los acumulados, sobre el total de piezas muestreadas hasta el momento

si $r \geq 1$ **entonces** // se ha observado una roja

DETENER; mostrar "Proceso NO ajustado"

Ajustar proceso; **ir** al inicio de **Fase de Cualificación**

si $h \geq 5$ **entonces** // se ha observado la 5.^a pieza amarilla

DETENER; mostrar "Proceso NO ajustado"

Ajustar proceso; **ir** al inicio de **Fase de Cualificación**

si $h \geq 3$ y $h - m \geq 2$ **entonces** // 3 amarillas y al menos 2 más que verdes

DETENER; mostrar "Proceso NO ajustado"

Ajustar proceso; **ir** al inicio de **Fase de Cualificación**

si $m - h \geq 2$ **entonces** // 2 verdes más que amarillas

DETENER; mostrar "Proceso ajustado"

Esperar hasta la próxima medición programada; **ir** al inicio de **Fase de Mantenimiento**

en otro caso

// seguir muestreando

ir a Paso 2

se tiene en cuenta la distinción entre las dos zonas amarillas. En dicha fase se sigue un esquema de muestreo en dos etapas: se muestrean 2 piezas, y si alguna es roja, el proceso debe detenerse y revisarse. Si ambas son verdes, el proceso se considera ajustado, y se deja de muestrear hasta la siguiente toma programada. En otro caso, se muestrean otras 3 piezas

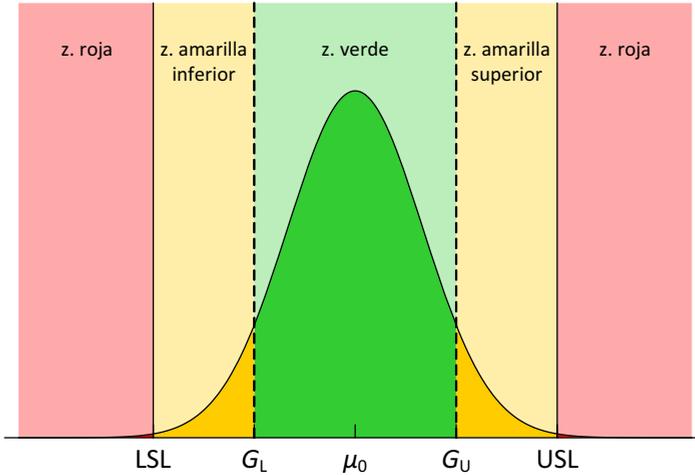


Figura 2.5: Método de clasificación del precontrol del cambio de media, para una característica de calidad con distribución normal, centrada y con capacidad potencial $C_p = 1$.

de una vez. Si en el total de piezas muestreadas se observan 3 o más en la zona amarilla inferior, o 3 o más en la zona amarilla superior, o bien alguna roja, el proceso necesita ser revisado. En otro caso, el proceso se considera ajustado (véase el algoritmo 2.5).

2.2.5. Precontrol simplificado

El precontrol simplificado es la tercera variación propuesta por Steiner (1997), que toma como base el esquema del precontrol en dos etapas. Sin embargo, en el precontrol simplificado las piezas son únicamente clasificadas en dos grupos: verde y amarillo. Los límites para la zona verde vienen dados en función de una constante c , de la manera siguiente:

$$G_L = \mu_0 - c \frac{USL - LSL}{2}, \quad (2.6)$$

$$G_U = \mu_0 + c \frac{USL - LSL}{2}. \quad (2.7)$$

Tomando $c = 0,5$ los límites de la zona verde coincidirían con los del

Algoritmo 2.5: Precontrol del cambio de media – Fase de mantenimiento

Paso 1 // Inicialización

$m = 0$ // número acumulado de verdes

$h_L = 0$ // número acumulado de piezas en la zona amarilla inferior

$h_U = 0$ // número acumulado de piezas en la zona amarilla superior

$r = 0$ // número acumulado de rojas

Paso 2

Extraer 2 piezas consecutivas

Actualizar m , h_L , h_U y r

si $m = 2$ **entonces** // las dos piezas son verdes

DETENER; mostrar "Proceso ajustado"

Esperar hasta la próxima medición programada; **ir** al inicio de **Fase de Mantenimiento**

si $r \geq 1$ **entonces** // alguna de las dos piezas es roja

DETENER; mostrar "Proceso NO ajustado"

Ajustar proceso; **ir** al inicio de **Fase de Cualificación**

en otro caso // 2 amarillas, o 1 verde y 1 amarilla

ir a Paso 3

Paso 3

Extraer 3 piezas

Actualizar m , h_L , h_U y r // Actualizamos los acumulados, sobre el total de 5 piezas muestreadas

si $r \geq 1$ **entonces** // alguna roja

DETENER; mostrar "Proceso NO ajustado"

Ajustar proceso; **ir** al inicio de **Fase de Cualificación**

si $h_L \geq 3$ o $h_U \geq 3$ **entonces** // 3 o más piezas en alguna de las dos zonas amarillas

DETENER; mostrar "Proceso NO ajustado"

Ajustar proceso; **ir** al inicio de **Fase de Cualificación**

en otro caso

DETENER; mostrar "Proceso ajustado"

Esperar hasta la próxima medición programada; **ir** al inicio de **Fase de Mantenimiento**

precontrol clásico y en dos etapas (véase la figura 2.6). En general, es necesaria una buena elección del valor de c , en función de la capacidad potencial del proceso y de qué desviación de la media del proceso sería considerada como inaceptable.

Steiner no especifica cómo adaptar la fase de cualificación a esta nueva forma de clasificar las unidades, en que no se considera la posibilidad de obtener una pieza roja.

En la fase de mantenimiento del precontrol simplificado se procede de la siguiente forma. Se muestrean 5 unidades consecutivas; si al menos 3 de ellas son amarillas, el proceso debe ser detenido y revisado. En otro caso, se considera que el proceso está ajustado (véase el algoritmo 2.6).

El procedimiento es más sencillo, aunque la cantidad esperada de piezas a muestrear es mayor que en el precontrol en dos etapas. La pérdida de eficiencia con respecto a éste es mínima, dado que, en procesos con una alta capacidad potencial, la probabilidad de muestrear una pieza roja (fuera de especificaciones) es muy pequeña.

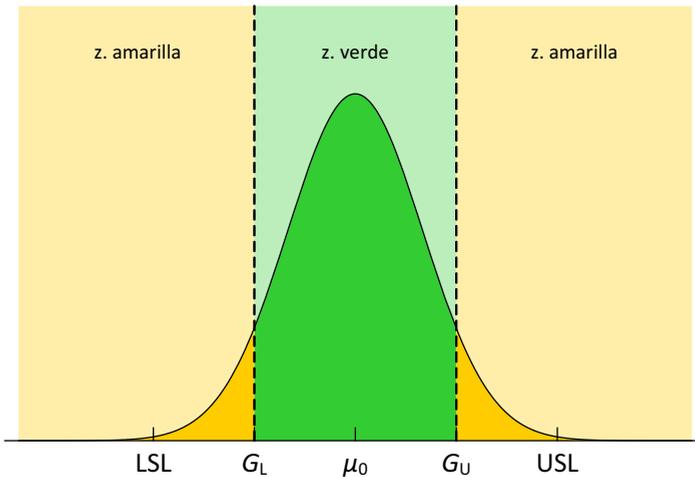


Figura 2.6: Método de clasificación del precontrol simplificado, con $c = 0,5$, para una característica de calidad con distribución normal, centrada y con capacidad potencial $C_p = 1$.

Algoritmo 2.6: Precontrol simplificado – Fase de mantenimiento

Paso 1 // Inicialización

$h = 0$ // número acumulado de amarillas

Paso 2

Extraer 5 piezas consecutivas

Actualizar h

si $h \geq 3$ entonces

DETENER; mostrar "Proceso NO ajustado"

Ajustar proceso; ir al inicio de **Fase de Cualificación**

en otro caso

DETENER; mostrar "Proceso ajustado"

Esperar hasta la próxima medición programada; ir al inicio de **Fase de Mantenimiento**

2.2.6. *Pairs*

Chamberlayne (1997) propone una modificación para la fase de cualificación del precontrol en un *technical report*.

La alternativa de Chamberlayne recibe el nombre de *pairs control chart*, o simplemente *pairs*, y consiste en modificar la anchura de la zona verde en función de la variabilidad natural (conocida o estimada) del proceso.

En concreto, Chamberlayne propone definir los límites de la zona verde de la siguiente forma:

$$G_L = \mu_0 - a\sigma_0, \quad (2.8)$$

$$G_U = \mu_0 + a\sigma_0, \quad (2.9)$$

siendo μ_0 y σ_0 , respectivamente, la media y la desviación típica del proceso cuando se encuentra bajo control y a un parámetro con valor comprendido entre 1,75 y 2. Por tanto, esta propuesta es muy similar al precontrol modificado de Gruska y Heaphy (1991) (véase la subsección

2.2.2), si bien Chamberlayne se refiere en su trabajo específicamente a la fase de cualificación del precontrol.

Cox et al. (2012) evalúa esta modificación frente al procedimiento de cualificación clásico del precontrol, y concluye que, cuando $C_p > 1,3333$, *pairs* es más sensible a la hora de detectar cambios en la media del proceso.

2.2.7. Precontrol modificado de Družovec et al.

En 1998, Družovec et al. presentaron otro precontrol modificado (MPC, en inglés), con el objeto, como todos las propuestas anteriores, de incrementar la sensibilidad ante variaciones del proceso y aumentar la eficiencia de la técnica, manteniendo su simplicidad.

El MPC incorpora modificaciones sobre el precontrol clásico tanto en la manera de clasificar las piezas muestreadas como en el diseño de la fase de cualificación.

Con la notación habitual $[G_L, G_U]$ para representar la zona verde, y denotando la zona amarilla como $[Y_L, Y_U] \setminus [G_L, G_U]$, Družovec et al. proponen definir dichas zonas de la siguiente forma:

$$\begin{aligned} G_L &= \frac{LSL + 2\mu_0}{3}, & Y_L &= \frac{2LSL + \mu_0}{3}, \\ G_U &= \frac{2\mu_0 + USL}{3}, & Y_U &= \frac{\mu_0 + 2USL}{3}. \end{aligned}$$

Es decir, en lugar de dividir el intervalo dentro de especificaciones en 4 partes iguales, constituyendo las dos centrales la zona verde y las dos laterales la zona amarilla —como se hace en la técnica clásica—, en el MPC se divide dicha zona en 6 intervalos de igual longitud. La zona verde ocupa, de nuevo, los dos intervalos centrales; la zona amarilla está formada por los dos intervalos contiguos a la zona verde; y los dos intervalos exteriores, junto con la zona fuera de especificaciones, constituyen la zona roja (véase la figura 2.7).

Por tanto, en el MPC, las piezas muestreadas cercanas a USL o LSL son consideradas como rojas, aunque estén dentro de las especificaciones.

En la fase de cualificación del MPC (véase el algoritmo 2.7), se

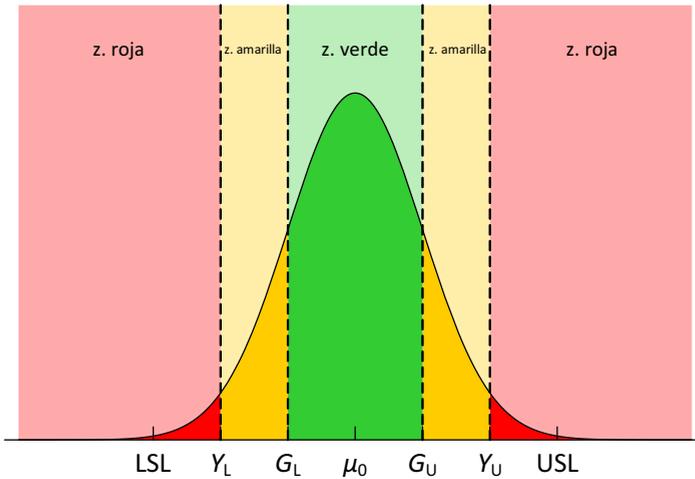


Figura 2.7: Método de clasificación del precontrol modificado de Družovec et al., para una característica de calidad con distribución normal, centrada y con capacidad potencial $C_p = 1$.

Algoritmo 2.7: Precontrol modificado de Družovec et al. – Fase de cualificación

Paso 1 // Inicialización

$h = 0$ // Número acumulado de amarillas

$r = 0$ // Número acumulado de rojas

Paso 2

Extraer 6 piezas consecutivas

Actualizar h y r

si $h \leq 2$ y $r = 0$ entonces

DETENER; mostrar "Proceso cualificado"

ir al inicio de Fase de Mantenimiento

en otro caso

DETENER; mostrar "Proceso NO cualificado"

Ajustar el proceso; ir al inicio de Fase de Cualificación

muestran siempre 6 piezas como máximo (a diferencia del resto de casos expuestos, en que el muestreo siempre era de tamaño variable). Si se obtienen no más de 2 piezas amarillas (no necesariamente consecutivas) y ninguna roja, el proceso es cualificado, y pasa a la fase de mantenimiento. En cualquier otro caso (más de 2 amarillas o alguna roja), el proceso se detiene para su revisión y pasar, posteriormente, de nuevo, a la fase de cualificación. La fase de mantenimiento del MPC es igual a la del precontrol clásico (algoritmo 2.2).

Según [Družovec et al.](#), el MPC mantiene las ventajas del precontrol y, en la fase de mantenimiento, reduce sensiblemente la probabilidad de aceptar un proceso descentrado. Su principal inconveniente es la alta de probabilidad de cualificar un proceso no capaz, en la fase de cualificación.

2.2.8. *Set-Up Process Algorithm*

En 2012, [Cox et al.](#) plantean una modificación de la fase de cualificación del precontrol, que en posteriores trabajos ([Cox et al., 2013b,a](#)) mejoran y validan. Dicha modificación recibe el nombre de *Set-Up Process Algorithm* (SUPA). En la subsección 2.3.4 nos referimos más extensamente a los trabajos de [Cox et al.](#) y al contexto en el que encuadran su propuesta. Aquí nos centraremos en describir la mejora que sugieren para el precontrol.

Tras constatar, como ya se ha mencionado aquí, que el precontrol clásico puede presentar tasas de falsa alarma elevadas en determinadas circunstancias debido a que su diseño no tiene en cuenta la variabilidad natural del proceso, [Cox et al.](#) proponen adaptar la anchura de la zona verde en función del índice de capacidad C_{pk} del proceso (1.3), con el fin de conseguir un procedimiento de precontrol que asegure una tasa de falsa alarma de 2%, valor que consideran aceptable ([Cox et al., 2013b](#); véase también la subsección 2.3.4).

Hasta donde hemos podido averiguar, la idea de modificar la anchura de la zona verde del precontrol aparece reflejada por primera vez en el precontrol modificado de [Gruska y Heaphy \(1991\)](#) que hemos revisado en la subsección 2.2.2, si bien son [San Matías et al. \(2004\)](#) los primeros

que proponen y desarrollan formalmente la determinación de su valor en función de la tasa de falsa alarma y la potencia que se desee para la fase de cualificación (véase más adelante la sección 2.4). Sin embargo, Cox et al. no citan este hecho, el cual resta originalidad a su aportación en lo que al precontrol se refiere. Curiosamente, los autores del SUPA sí citan directamente el trabajo de San Matías et al. como fuente del cálculo de la probabilidad de falsa alarma.

Por último, cabe mencionar que Cox et al. no modifican el número de piezas verdes o amarillas consecutivas que es necesario muestrear para terminar el procedimiento. Con ello se quiere mantener la ventaja del precontrol frente a otros métodos, tales como los gráficos de aceptación, permitiendo la posibilidad descartar un proceso con sólo muestrear 2 piezas.

En definitiva, el SUPA es presentado como un método que consta de los siguientes pasos:

- 1.º) determinar el valor de la anchura de la zona verde en función del mínimo valor deseado para C_{pk} ;
- 2.º) aplicar la fase de cualificación del precontrol de acuerdo con el esquema clásico (algoritmo 2.1), teniendo en cuenta las definiciones modificadas de las zonas verde y amarilla.

Cox et al. (2013b,a) comparan el comportamiento del SUPA frente a otras propuestas. Las conclusiones pueden consultarse en la subsección 2.3.4.

2.2.9. Resumen de las variantes del precontrol

Hemos descrito 8 propuestas de mejora del precontrol clásico, de las cuales 5 se centran en la fase de mantenimiento y 3 en la fase de cualificación.

Resumiendo lo ya expuesto, y basándonos en lo dicho por Steiner (1997), puede decirse que, en la fase de mantenimiento, el precontrol de las diez unidades está indicado siempre que el muestreo de más piezas

no resulte caro o costoso. Por otro lado, el precontrol del cambio de media resulta más eficaz que el precontrol en dos etapas para detectar cambios en la media del proceso. Por último, el precontrol simplificado proporciona resultados similares al precontrol en dos etapas, y está especialmente indicado en procesos con alta capacidad potencial (es decir, procesos cuya desviación típica es pequeña en comparación con los límites de especificación), siempre que se ajuste convenientemente su parámetro c .

De las propuestas que abordan la mejora de la fase de mantenimiento del precontrol, la de [Cox et al. \(2012\)](#) nos parece más completa y potente que la de [Družovec et al. \(1998\)](#), ya que la liberación de la anchura de la zona verde permite adaptar el comportamiento de la técnica a más situaciones. De hecho, como se ha mencionado anteriormente, dicha flexibilización del precontrol fue ya propuesta en 2004 por [San Matías et al.](#), con resultados muy interesantes, como se detallará posteriormente en la sección 2.4. Por otro lado, la propuesta de [Chamberlayne \(1997\)](#), *pairs*, que también liga la anchura de la zona verde a la variabilidad natural del proceso, es más *rígida* que el SUPA de [Cox et al.](#), que, de hecho, la supera (véase más adelante en la subsección 2.3.4).

2.3. Otras contribuciones recientes

Apenas se encuentran en la literatura análisis o propuestas de mejora relacionados con el precontrol, además de los publicados por [Steiner \(1997\)](#), [Družovec et al. \(1998\)](#) y [San Matías et al. \(2004\)](#). Únicamente nos constan los siguientes:

- un trabajo de [Olorunniwo y Knight \(1995\)](#) en el que se proponen sendas metodologías para estimar los parámetros μ y σ de un proceso monitorizado mediante precontrol;
- un caso de estudio publicado por [Vermani \(2000\)](#) en el que aporta una manera de monitorizar varias características de calidad en un mismo gráfico de precontrol;

- una hibridación del precontrol con los gráficos de control por zonas (véase la subsección 1.2.2) propuesta por [Torbeck \(2005\)](#);
- una aplicación y mejora del precontrol como técnica para el control de procesos con lotes pequeños y en los que la puesta a punto es especialmente importante ([Cox et al., 2012, 2013b,a](#); véase también la subsección 2.2.8); y
- un estudio de [Pan \(2007\)](#) sobre el precontrol multivariante, basado en un trabajo previo de [Hubele \(1988\)](#). Hasta donde hemos podido averiguar, *no* se han publicado otros intentos de extensión del precontrol al caso multidimensional.

A diferencia de las descritas en la sección 2.2, las propuestas de [Olorunniwo y Knight](#), [Vermani](#), [Hubele](#) y [Pan](#) no persiguen *mejorar* la técnica clásica mediante variaciones en el diseño de todos o algunos de sus tres elementos (método de clasificación, fase de cualificación, fase de mantenimiento), sino que plantean *adaptaciones* de la misma a unos determinados contextos o necesidades.

[Torbeck](#) propone un gráfico que incorpora ideas del precontrol y de los gráficos por zonas. Según nuestro criterio, no puede considerarse una variante del precontrol, sino una nueva técnica surgida a partir de éste.

Por otro lado, continuamos analizando la propuesta de [Cox et al.](#) (véase la subsección 2.2.8), haciendo aquí hincapié en su ámbito de aplicación.

Con todo ello completamos el *mapa* del estado actual del precontrol, restando sólo por analizar el trabajo ya mencionado de [San Matías et al. \(2004\)](#), que abordaremos en las secciones 2.4 y siguientes, y que será el punto de partida de nuestra propia propuesta.

2.3.1. Monitorización de parámetros del proceso con precontrol

[Olorunniwo y Knight \(1995\)](#) proponen subdividir el rango de tolerancias [LSL, USL] en N intervalos o *subzonas* de igual anchura, con N igual a 4, 6 u 8, y siendo $N = 8$ el valor recomendado por los autores. Después,

plantean utilizar las sucesivas observaciones clasificadas en subzonas obtenidas en la fase de mantenimiento del precontrol para estimar por un lado la media del proceso μ y por otro lado la desviación típica σ . Su propuesta intenta mantener en todo momento la simplicidad del precontrol, en el sentido de que no utiliza las observaciones numéricas concretas, sino su clasificación en subzonas.

Olorunniwo y Knight llevan a cabo una calibración de los parámetros involucrados en cada uno de los dos métodos que proponen —para estimar μ y σ , respectivamente— mediante un estudio de análisis de la varianza, y comparan el rendimiento de sus estimaciones frente a las clásicas, basadas en gráficos de control, resultando bastante competitivas. Finalmente, dan recomendaciones para la aplicación de sus métodos *a pie de línea*.

2.3.2. Precontrol sobre el porcentaje de tolerancias

Vermani (2000) propone, para cada característica de calidad de interés X , monitorizar

$$X^* = \frac{X - \mu_0}{(USL - LSL)/2},$$

siendo μ_0 el valor nominal para X , el cual, como habitualmente, se supondrá centrado en el intervalo de tolerancias $[LSL, USL]$. El valor de X^* representa, por tanto, la diferencia entre X y su valor objetivo μ_0 , medida en relación a la distancia de μ_0 a los límites de tolerancia.

Una vez *estandarizada* de esta forma la variable X , la propuesta de Vermani consiste en aplicar el procedimiento de precontrol clásico, representando los valores observados de X^* (normalmente expresados en forma de porcentaje) en un gráfico, esta vez *sí* a escala. Se sugiere realizar con ayuda de una hoja de cálculo todo el proceso de normalización y representación gráfica.

La ventaja que ofrece este método es que permite representar en un único gráfico tantas características de calidad como se desee (tanto correspondientes a un único producto como a diferentes productos procedentes de una misma línea), en lugar de mantener varios gráficos, y

visualizar así fácilmente si la producción en general se mantiene dentro de tolerancias. A cambio, obliga al operario a realizar cálculos durante o inmediatamente después de la toma de datos.

2.3.3. Gráfico de sectores

Torbeck (2005) propone una nueva herramienta llamada gráfico de sectores (en inglés, *sector chart*), tomando ideas del precontrol y de los gráficos de control por zonas (véase la subsección 1.2.2).

Torbeck plantea, al estilo de los gráficos de zonas, dividir el rango de tolerancias en un cierto número de tramos de igual o distinta longitud; desde 4 hasta 8 o 10, dependiendo de los casos. Dichos tramos, junto con las zonas fuera de tolerancias, son denominados *sectores*. Cada sector mantiene el *color* (verde, amarillo o rojo) que le correspondería según el procedimiento de precontrol, a efectos visuales, que no de clasificación de las observaciones.

Siguiendo también el esquema de los gráficos de zonas, a cada sector se le asigna una puntuación, creciente según se aleja del valor central. Dicha puntuación puede ser ajustada en función de cómo de sensible se quiere que sea el gráfico a cambios en la media del proceso. Las reglas de decisión son prácticamente las mismas que las de los gráficos de zonas: cada observación recibe una puntuación en función del sector en el que es ubicada, la cual se suma a la puntuación acumulada hasta ese momento, excepto si la observación cae dentro de los sectores verdes o bien está en el lado opuesto de la línea central respecto de la observación anterior, caso en el que la puntuación se reinicia. En el momento en que la puntuación acumulada supera un cierto umbral, que suele coincidir con el valor asignado a los sectores rojos, se desencadena una señal de alarma.

Así, el gráfico de sectores continúa poniendo el foco en las especificaciones y no tanto en la variabilidad natural del proceso, como el precontrol, pero sustituye los mecanismos de clasificación y decisión de éste por los de los gráficos de control por zonas. Aun así, como se desprende de su mecanismo, está principalmente orientado a la detección de cambios en la media del proceso, y no tanto en la dispersión.

El autor destaca las bondades de su propuesta (casi todas ellas compartidas con el precontrol, tales como la sencillez y la rapidez de uso, la no necesidad de cálculos ni de asunción de ninguna hipótesis acerca de la distribución de los datos, etc.), pero no aporta datos de ninguna experiencia numérica ni comparaciones en términos de eficiencia con el precontrol ni con otros gráficos.

2.3.4. Aplicación del SUPA a la puesta a punto de procesos

En la subsección 2.2.8 se ha descrito una modificación de la fase de cualificación del precontrol propuesta por Cox et al. (2012, 2013b,a), consistente en recalcular la anchura de la zona verde en función de la tasa de falsa alarma deseada. Dicho método recibe el nombre de SUPA, por *Set-Up Process Algorithm*.

La versión inicial del SUPA presentada por Cox et al. (2012) es un poco más compleja, pues realiza una hibridación entre el precontrol y los gráficos de aceptación (concretamente, tras obtener la cualificación del proceso mediante el precontrol, se pasa un filtro adicional consistente en exigir que la media estimada del proceso esté dentro de los límites definidos por un gráfico de control de aceptación; véase la subsección 2.1.4), pero ese detalle desaparece en las versiones definitivas de Cox et al. (2013b,a).

Los autores se centran en el papel que puede desempeñar la fase de cualificación del precontrol en la validación de la puesta a punto de procesos con lotes de tamaño pequeño y muy espaciados en el tiempo, y en los que la adecuación a unas tolerancias dadas sea especialmente importante. En tales procesos, la puesta a punto antes de la producción de cada lote es fundamental, ya que, debido a la demora entre series de producción, las causas especiales de variabilidad más importantes se producen *entre lotes*. Este tipo de situaciones se da, por ejemplo, en la producción de piezas destinadas a recambios: pueden estar produciéndose en pequeñas cantidades a lo largo de muchos años, con bastante distancia temporal entre lotes; y, para garantizar su utilidad, es necesario asegurar que las piezas cumplen con las especificaciones de diseño (Cox et al.,

2013a). Los autores se refieren a esta clase de procesos como *set-up dominant processes*³.

Concretamente, Cox et al. redefinen la anchura de la zona verde de tal forma que, dado un valor deseado para el índice de capacidad C_{pk} del proceso, la tasa de falsa alarma de la fase de cualificación del precontrol para un proceso con una capacidad igual o superior a dicho valor sea igual o inferior al 2 %, valor que, citando a Ledolter y Swersey (1997), los autores consideran apropiado para el tipo de procesos objeto de interés. A tal efecto, proporcionan una tabla de búsqueda con la anchura de la zona verde calculada en función del valor mínimo deseado para C_{pk} .

Cox et al. especifican que se refieren a dos tasas de falsa alarma que tratan de acotar simultáneamente: la probabilidad de no cualificar un proceso centrado, y la de no cualificar un proceso capaz (véanse más adelante las subsecciones 2.5.1 y 2.5.2). En realidad, se basan en la fórmula que da la probabilidad de cualificar un proceso deducida por San Matías et al. (2004) para la fase de cualificación del precontrol clásico, pero no detallan exactamente el modo de calcular la anchura de la zona verde a partir de C_{pk} .

Además, Cox et al. (2013b) destacan que el SUPA es un método no paramétrico, propiedad que —afirman— hereda del precontrol clásico. Sin embargo, su principal novedad respecto a éste consiste en asegurar una cierta tasa de falsa alarma, propiedad que es inherente a su diseño, y esto sólo se puede hacer bajo la hipótesis de que el proceso sigue alguna distribución de probabilidad; supuestamente, la distribución normal, aunque los autores no la mencionan a la hora de explicar el cálculo de la anchura de la zona verde.

Cox et al. (2012, 2013b) validan el comportamiento de su propuesta frente al precontrol clásico, los gráficos de control de aceptación (véanse las subsecciones 1.2.2, *pairs* (véase la subsección 2.2.6) y 2.1.4) y el gráfico \bar{X} - R para pequeños lotes de Pillet (1996) (véase la subsección 1.2.2), mediante simulación, suponiendo una distribución normal y para

³Los procesos de estampación y los de moldeo por inyección son dos ejemplos de este tipo de procesos.

dos escenarios: proceso centrado y no capaz, y proceso no centrado y capaz. Posteriormente, [Cox et al. \(2013a\)](#) amplían la simulación a otras distribuciones de probabilidad, pero sólo comparando el SUPA con el gráfico de [Pillet](#).

Las principales conclusiones son que el SUPA es tanto o más sensible que el resto de alternativas probadas a la hora de detectar situaciones no deseadas, y que esto es a costa de requerir, por término medio, muestrear más unidades. Los autores del SUPA atribuyen, además, a su método las ventajas ya conocidas propias del precontrol.

2.3.5. Precontrol multivariante

El primer trabajo publicado que aborda una versión multivariante del precontrol es debido a [Hubele \(1988\)](#). En él se estudia la extensión de las técnicas de calibración de límites estrechos ([Dudding y Jennett, 1944](#); véase también la nota al pie de la página 49) al caso multivariante y, en concreto, se diseña un procedimiento de precontrol para el caso en que la característica de calidad de interés presente una distribución normal multivariante $\mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\mathbf{X} \in \mathbb{R}^p$, $p \geq 2$.

En su propuesta, [Hubele](#) extiende de manera natural el criterio de clasificación de las observaciones en verdes, amarillas y rojas al caso de una distribución normal multivariante, apoyándose en las propiedades del estadístico $T^2 = (\mathbf{X} - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1} (\mathbf{X} - \boldsymbol{\mu}_0)$ de [Hotelling \(1931, 1947\)](#), siendo $\boldsymbol{\mu}_0$ y $\boldsymbol{\Sigma}_0$ los valores nominales para la media y la matriz de varianzas-covarianzas del proceso, respectivamente.

Concretamente, utilizando que T^2 sigue una distribución χ^2 con p grados de libertad cuando $\mathbf{X} \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ y suponiendo que las especificaciones para \mathbf{X} vienen dadas a partir de la distribución de probabilidad, se definen las zonas verde, amarilla y roja de la manera siguiente:

$$\begin{aligned} \text{Z. verde} &= \left\{ \mathbf{x} \in \mathbb{R}^p \mid (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq \chi_p^2(\alpha_G) \right\}, \\ \text{Z. amarilla} &= \left\{ \mathbf{x} \in \mathbb{R}^p \mid \chi_p^2(\alpha_G) < (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq \chi_p^2(\alpha_S) \right\} \quad \text{y} \end{aligned}$$

$$Z. \text{ roja} = \left\{ \mathbf{x} \in \mathbb{R}^p \mid (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) > \chi_p^2(\alpha_S) \right\},$$

donde α_S representa la probabilidad de que una pieza esté fuera de especificaciones, y α_G la probabilidad de que una pieza esté fuera de la zona verde, cuando el proceso se encuentra bajo control ($\alpha_G \geq \alpha_S$); ambas probabilidades serían establecidas de antemano. Asimismo, $\chi_p^2(\alpha)$ representa, como es habitual, el valor que deja a su derecha una probabilidad de α en una variable aleatoria χ^2 con p grados de libertad, es decir, $\Pr(\chi_p^2 \geq \chi_p^2(\alpha)) = \alpha$. Como es sabido, las regiones definidas de esta forma están delimitadas por elipsoides (véase la figura 2.8). Es decir, en el fondo se está asumiendo que las tolerancias sobre \mathbf{X} se pueden traducir o expresar geoméricamente en forma de elipsoide centrado en $\boldsymbol{\mu}_0$ (de manera similar a la definición de los límites de control que se maneja en el gráfico T^2 de [Hotelling](#)).

El valor de α_S puede determinarse a partir de experimentación previa, y el de α_G puede asignarse arbitrariamente. [Hubele](#) utiliza en sus experiencias numéricas $\alpha_S = 0,001$ (correspondería a un proceso potencialmente muy capaz) y $\alpha_G = 1 - 0,8664 = 0,1336$, que coincide con la proporción estimada de piezas que caerían fuera de la zona verde en el precontrol clásico unidimensional, en un proceso con capacidad potencial $C_p = 1$.

Para el caso de una característica de calidad bidimensional ($p = 2$), [Hubele](#) propone dividir el espacio de resultados posibles en 9 regiones (véase la figura 2.8), que se superpondrían a las zonas verde, amarilla y roja arriba descritas, con el fin de identificar diferentes *subtipos* de piezas amarillas y rojas.

La autora no se refiere de manera explícita a las fases de cualificación y mantenimiento del precontrol; simplemente desarrolla reglas a seguir para ajustar el proceso cuando dos o más piezas amarillas consecutivas son muestreadas, en función de en cuál o cuáles de las 9 subregiones son localizadas.

Las experiencias numéricas que desarrolla le permiten concluir que este tipo de enfoque multidimensional es superior en algunos casos a

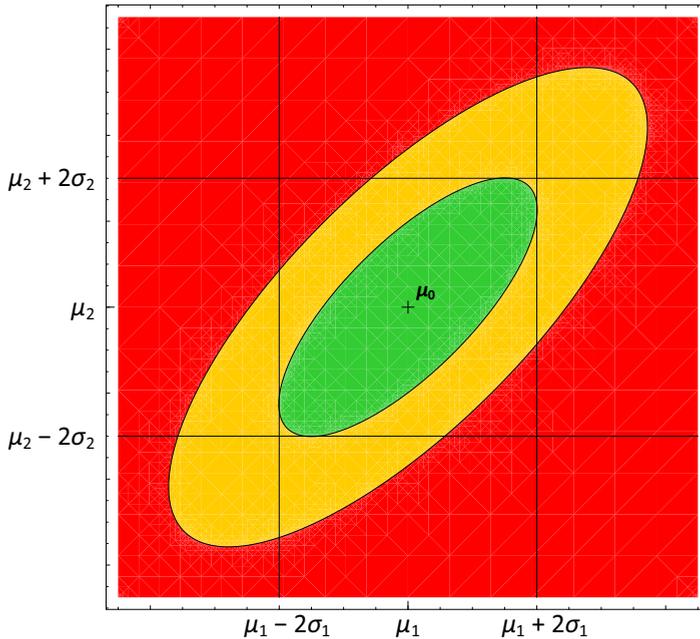


Figura 2.8: Zonas verde, amarilla y roja en el precontrol multivariante de Hubele (1988), con probabilidades de verde y roja igual a 0,8664 y 0,001, respectivamente, para el caso de una característica de calidad normal bivalente de media $\mu_0 = (\mu_1, \mu_2)$ y coeficiente de correlación $\rho = 0,75$. Elaboración propia a partir una figura de Hubele (1988).

realizar un precontrol para cada característica de calidad por separado.

Posteriormente, Pan (2007) retoma el trabajo de Hubele y explora diferentes mejoras en el diseño de las fases de cualificación y mantenimiento. Por un lado, Pan plantea fijar el tamaño muestral de la fase de cualificación a 10 unidades para mejorar su comportamiento.

Por otro lado, tras evaluar diferentes reglas de decisión para la fase de mantenimiento, el autor propone la siguiente: muestrear 3 piezas; si se obtienen al menos 2 verdes y ninguna roja, el proceso se considera ajustado; si se obtienen 3 amarillas o alguna roja, se considera que el proceso necesita ser revisado; en otro caso, se muestrean dos piezas más para tomar la decisión. Pan también sugiere aumentar el tamaño de la

zona verde propuesto por Hubele, con el fin de reducir la tasa de falsa alarma en la fase de mantenimiento, en función de los requerimientos del usuario.

Por último, Pan compara el comportamiento de su propuesta frente a los gráficos T^2 de Hotelling (1947). Concluye que, en término medio, estos últimos tardan menos en detectar que un proceso está fuera de control durante la fase de mantenimiento, y que aun así el precontrol sigue resultando una alternativa interesante, sobre todo en la fase de cualificación para valorar la capacidad del proceso, dada su simplicidad.

2.4. Precontrol generalizado

En las secciones anteriores hemos descrito la técnica del precontrol clásico, así como algunas de las alternativas a la misma que se han venido planteando en la literatura.

Como se ha visto, casi todas las variantes del precontrol expuestas en la sección 2.2 se diferencian principalmente en el esquema de su fase de mantenimiento o en la manera de clasificar las piezas, permaneciendo invariable el diseño de la fase de cualificación (a excepción del MPC de Družovec et al., 1998 y del SUPA de Cox et al., 2013b).

San Matías et al. (2004) realizaron otra propuesta, cuyo objeto es mejorar el comportamiento de la técnica clásica en la fase de cualificación. Consiste en flexibilizar la elección de los parámetros siguientes:

- W_G , la anchura de la zona verde,
- k , el número de unidades verdes consecutivas necesarias para cualificar el proceso, y
- t , el número de unidades amarillas consecutivas necesarias para detener y ajustar el proceso.

Como sabemos, en el precontrol clásico estos tres parámetros W_G , k y t están fijados y toman los valores $\frac{USL-LSL}{2}$, 5 y 2, respectivamente. La propuesta de San Matías et al. consiste en liberar dichos parámetros, eligiendo para ellos el valor que más convenga en cada situación. Por

ello, denominaremos esta alternativa a la técnica clásica como *precontrol generalizado*⁴.

La elección del parámetro W_G afecta a la definición de las zonas verde y amarilla. Los valores de k y t influyen directamente en el diseño del procedimiento de cualificación, primera fase del precontrol.

Así, la fase de cualificación del precontrol generalizado queda como sigue. Se eligen convenientemente los parámetros W_G , k y t . A continuación, se muestrean sucesivas piezas hasta encontrar k verdes consecutivas (y en tal caso se cualifica el proceso), o bien hasta encontrar 1 roja o t amarillas consecutivas (en cuyo caso el proceso necesita ser detenido y revisado). El algoritmo 2.8 refleja el esquema que acabamos de describir.

San Matías et al. demuestran que una correcta elección de los parámetros de la fase de cualificación puede contribuir a aumentar la potencia del precontrol para detectar desviaciones del proceso, y disminuir la tasa de falsa alarma, mejorando de esta manera el comportamiento de la técnica clásica. De hecho, proponen un sencillo algoritmo para la elección de dichos parámetros, de manera que se verifiquen los requerimientos del usuario en dichos términos.

Tanto ahora como en las secciones anteriores nos hemos referido a la bondad de un esquema de precontrol en términos de su capacidad para detectar desviaciones del proceso. A continuación, detallaremos con mayor propiedad de qué manera puede medirse la potencia y la eficiencia de un plan de precontrol dado, centrándonos en la fase de cualificación del precontrol generalizado.

⁴En su trabajo, San Matías et al. no utilizan ningún nombre en especial para referirse a su propuesta.

2.5. Evaluación del comportamiento de los planes de precontrol⁵ generalizados

En adelante, llamaremos *plan de precontrol* a un esquema de precontrol generalizado en el que ya han sido fijados los valores de los parámetros W_G , k y t . Dado un plan de precontrol, es fundamental poder valorar su comportamiento y la fiabilidad de su resultado, es decir, si es una herramienta útil para saber realmente cuándo el proceso está ajustado y cuándo necesita ser revisado.

En otras palabras, es necesario cuantificar su capacidad para detectar situaciones de desajuste que provoquen una alta proporción de piezas fuera de especificaciones, como pueden ser una desviación inaceptable de la media o una excesiva variabilidad del proceso.

A continuación exponemos las dos formas más habituales de evaluar la significación y la potencia de un plan de precontrol. En primer lugar, se considera el plan de precontrol como un contraste de hipótesis sobre la media real del proceso; posteriormente, lo evaluamos cuando éste es utilizado como un test sobre la capacidad potencial del mismo.

2.5.1. Precontrol como contraste sobre la media

En este apartado describiremos la manera habitual en que se evalúa el comportamiento de un plan de precontrol, esto es, en términos de su potencia para detectar desviaciones de la media del proceso respecto del valor objetivo o nominal.

Concretamente, el precontrol puede ser visto como un procedimiento para resolver el siguiente contraste de hipótesis:

$$\begin{cases} H_0 : \delta = 0 \\ H_1 : \delta > 0 \end{cases} \quad (2.10)$$

⁵Nuestro trabajo se ha centrado únicamente en la fase de cualificación del precontrol. Por este motivo, a partir de aquí usaremos el nombre de precontrol para referirnos a dicha fase.

Algoritmo 2.8: Precontrol generalizado – Fase de cualificación

Paso 1 // InicializaciónElegir los valores de W_G , k y t $m = 0$ // Número de verdes consecutivas $h = 0$ // Número de amarillas consecutivas**Paso 2** // Muestreo intensivo

Extraer una pieza

si la pieza es verde **entonces** $m = m + 1$ $h = 0$ **ir a Paso 3****si** la pieza es amarilla **entonces** $m = 0$ $h = h + 1$ **ir a Paso 3****si** la pieza es roja **entonces** $m = 0$ $h = 0$ **DETENER; mostrar** "Proceso NO cualificado"Ajustar el proceso; **ir** al inicio de **Fase de Cualificación****Paso 3****si** $m < k$ y $h < t$ **entonces**

// Seguir muestreando

ir a Paso 2**en otro caso****ir a Paso 4****Paso 4****si** $m = k$ **entonces** // k verdes consecutivas**DETENER; mostrar** "Proceso cualificado"**ir** al inicio de **Fase de Mantenimiento****en otro caso** // t amarillas consecutivas**DETENER; mostrar** "Proceso NO cualificado"Ajustar el proceso; **ir** al inicio de **Fase de Cualificación**

donde δ es la desviación estandarizada en valor absoluto de la media real del proceso μ respecto del valor nominal o valor objetivo μ_0 , es decir:

$$\delta = \frac{|\mu_0 - \mu|}{\sigma_0}. \quad (2.11)$$

Si el precontrol termina cualificando el proceso, se toma como cierta la hipótesis nula (es decir, no se puede rechazar que el proceso esté centrado, y por tanto que la desviación del proceso δ sea nula); por el contrario, si el proceso no es cualificado, se aceptará que existe una diferencia significativa entre el valor real de la media del proceso μ y el valor nominal μ_0 .

Con esta metodología, la probabilidad de no rechazar H_0 es la probabilidad de cualificar el proceso, cuya expresión (San Matías et al., 2004) es la siguiente:

$$\Pr(\text{cualificar} \mid \delta) = p_g(\delta)^k \frac{\sum_{j=0}^{t-1} p_y(\delta)^j}{1 - \sum_{j=1}^{t-1} p_y(\delta)^j \sum_{i=1}^{k-1} p_g(\delta)^i}, \quad (2.12)$$

siendo $p_g(\delta)$ la probabilidad de extraer una pieza verde y $p_y(\delta)$ la de una amarilla, para un proceso con una cierta desviación δ ⁶.

⁶Este cálculo tiene en cuenta, entre otras cosas, que el tamaño muestral de la fase de cualificación del precontrol es variable, que ésta no termina hasta que se da por cualificado o no cualificado el proceso, y que el evento «cualificar» (observar k verdes consecutivas) puede suceder en cualquier momento del muestreo (véase el algoritmo 2.8). Esto es aplicable también al precontrol clásico (con $k = 5$, lógicamente; véase la subsección 2.1.2) y a la mayoría de propuestas alternativas que se han presentado en la sección 2.2.

Sin embargo, algunos autores (por ejemplo, Tang y Barnett, 1994, Ledolter y Swersey, 1997 y Družovec et al., 1998) simplifican el cálculo de las probabilidades asociadas al precontrol, afirmando, por ejemplo, que la probabilidad de que un proceso termine siendo cualificado es sencillamente $p_g(\delta)^k$ (utilizando nuestra notación), es decir, como si encontrar una sola pieza amarilla obligase a detener y no cualificar el proceso, o equivalentemente, como si el tamaño muestral de esta fase del precontrol estuviera limitado a k piezas.

De hecho, Shainin y Shainin (1989) lanzan algunas críticas a los que se basan en ese tipo de cálculos erróneos para justificar la necesidad de modificaciones al precontrol.

Dado que en capítulos posteriores haremos uso de la expresión (2.12), deducida por San Matías et al. (2004), para desarrollar nuestra propuesta en torno al precontrol, comprobamos a continuación que dicha expresión está bien definida para cualquier valor de los parámetros involucrados.

Proposición 2.1 (Buena definición de $\Pr(\text{cualificar} \mid \delta)$). *La expresión para la probabilidad de que el precontrol termine cualificando un proceso dado (2.12) está bien definida para cualquier valor posible de los parámetros k, t, W_G, δ y C_p , esto es, para cualesquiera valores $k, t \geq 1$ enteros, $W_G > 0, \delta > 0$ y $C_p > 0$.*

Demostración. Para el caso en que $k = 1$ o $t = 1$, puede comprobarse a partir del desarrollo realizado por San Matías et al. (2004) que

$$\Pr(\text{cualificar} \mid \delta, k = 1 \text{ o } t = 1) = p_g(\delta)^k \sum_{j=0}^{t-1} p_y(\delta)^j. \quad (2.13)$$

Esta última expresión puede obtenerse igualmente a partir de (2.12) utilizando

$$\sum_{i=1}^{m-1} r^i = \sum_{i=0}^{m-1} r^i - 1. \quad (2.14)$$

Por tanto, puede considerarse válida la expresión (2.12) también para el caso en que $k = 1$ o $t = 1$.

A continuación vamos a comprobar que esta expresión para $\Pr(\text{cualificar} \mid \delta)$ está bien definida, es decir, que el denominador de (2.12) siempre es mayor que cero y el resultado del cociente siempre es menor que 1, para cualquier valor posible de los parámetros k, t, W_G, δ y C_p . En efecto, si tenemos en cuenta que el resultado de una serie geométrica con $|r| < 1$ es

Nótese que, por ejemplo, aplicando el precontrol clásico —es decir, $W_G = (USL - LSL)/2$, $k = 5$ y $t = 2$ — sobre un proceso centrado con capacidad potencial $C_p = 1$ y suponiendo normalidad, se tiene que $p_g(0)^5 = 48,81\%$, mientras que la probabilidad de cualificar el proceso según (2.12) sería $87,71\%$ (casi un 80% mayor).

En el presente trabajo hemos optado por seguir la interpretación realizada por San Matías et al. (2004), por ser, a nuestro entender, más exacta y fiel a la idea original.

$$\sum_{i=0}^{+\infty} r^i = \frac{1}{1-r}, \quad (2.15)$$

entonces, usando (2.14) y (2.15), y teniendo en cuenta que, por definición, las probabilidades p_g y p_y no son nunca negativas, se tiene:

$$\begin{aligned} 1 - \sum_{j=1}^{t-1} p_y(\delta)^j \sum_{i=1}^{k-1} p_g(\delta)^i &= 1 - \left(\sum_{j=0}^{t-1} p_y(\delta)^j - 1 \right) \left(\sum_{i=0}^{k-1} p_g(\delta)^i - 1 \right) \\ &> 1 - \left(\frac{1}{1-p_y(\delta)} - 1 \right) \left(\frac{1}{1-p_g(\delta)} - 1 \right) \end{aligned} \quad (2.16)$$

$$= 1 - \frac{p_g(\delta)p_y(\delta)}{(1-p_g(\delta))(1-p_y(\delta))}. \quad (2.17)$$

Veamos si (2.17) es positivo:

$$\begin{aligned} 1 - \frac{p_g(\delta)p_y(\delta)}{(1-p_g(\delta))(1-p_y(\delta))} > 0 &\Leftrightarrow (1-p_g(\delta))(1-p_y(\delta)) > p_g(\delta)p_y(\delta) \\ &\Leftrightarrow 1 - p_g(\delta) - p_y(\delta) + p_g(\delta)p_y(\delta) \\ &> p_g(\delta)p_y(\delta) \\ &\Leftrightarrow 1 > p_g(\delta) + p_y(\delta). \end{aligned} \quad (2.18)$$

La expresión (2.18) a la que hemos llegado es cierta, ya que, por su definición, la suma de las probabilidades $p_g(\delta)$ y $p_y(\delta)$ representa la probabilidad de que una observación esté dentro del rango de especificaciones [LSL, USL], de modo que ésta siempre es menor que 1. Por tanto, podemos concluir que (2.17) es positivo y, en consecuencia, también lo es el denominador de (2.12).

Asimismo, podemos probar que la expresión (2.12) es inferior a 1 en cualquier caso. Efectivamente, partiendo de (2.12) y usando de nuevo

(2.14) y (2.15) sobre la sumas parciales $\sum_{j=0}^{t-1} p_y(\delta)^j$ y $\sum_{j=1}^{t-1} p_y(\delta)^j$, tenemos:

$$\begin{aligned} \Pr(\text{cualificar} \mid \delta) &= p_g(\delta)^k \frac{\sum_{j=0}^{t-1} p_y(\delta)^j}{1 - \sum_{j=1}^{t-1} p_y(\delta)^j \sum_{i=1}^{k-1} p_g(\delta)^i} \\ &< \frac{p_g(\delta)^k \frac{1}{1 - p_y(\delta)}}{1 - \left(\frac{1}{1 - p_y(\delta)} - 1 \right) \sum_{i=1}^{k-1} p_g(\delta)^i}. \end{aligned} \quad (2.19)$$

Como se ha dicho arriba, $p_g(\delta) + p_y(\delta) < 1$, es decir, $1 - p_y(\delta) > p_g(\delta)$. Aplicando esto a (2.19) obtenemos lo siguiente:

$$\begin{aligned} \Pr(\text{cualificar} \mid \delta) &< \frac{p_g(\delta)^k \frac{1}{p_g(\delta)}}{1 - \left(\frac{1}{p_g(\delta)} - 1 \right) \sum_{i=1}^{k-1} p_g(\delta)^i} \\ &= \frac{p_g(\delta)^{k-1}}{\left(1 + \sum_{i=1}^{k-1} p_g(\delta)^i \right) - \frac{1}{p_g(\delta)} \sum_{i=1}^{k-1} p_g(\delta)^i} \\ &= \frac{p_g(\delta)^{k-1}}{\sum_{i=0}^{k-1} p_g(\delta)^i - \sum_{i=0}^{k-2} p_g(\delta)^i} = \frac{p_g(\delta)^{k-1}}{p_g(\delta)^{k-1}} = 1, \end{aligned}$$

como se quería demostrar.

En el caso en que $k = 1$, podemos usar la expresión particular (2.13) y de nuevo las relaciones (2.14), (2.15) y $p_g(\delta) + p_y(\delta) < 1$ para concluir también que

$$\Pr(\text{cualificar} \mid \delta, k = 1) = p_g(\delta) \sum_{j=0}^{t-1} p_y(\delta)^j < \frac{p_g(\delta)}{1 - p_y(\delta)} < \frac{p_g(\delta)}{p_g(\delta)} = 1,$$

como queríamos probar.

Estas demostraciones son válidas siempre que podamos suponer que la probabilidad de obtener una pieza verde $p_g(\delta)$ no se anula en ningún momento, cosa que, como se verá un poco más adelante, es perfectamente asumible. \square

Según (2.12), la probabilidad de error tipo I para este contraste será:

$$\alpha = 1 - \Pr(\text{cualificar} \mid \delta = 0) = 1 - p_g(0)^k \frac{\sum_{j=0}^{t-1} p_y(0)^j}{1 - \sum_{j=1}^{t-1} p_y(0)^j \sum_{i=1}^{k-1} p_g(0)^i}. \quad (2.20)$$

Si suponemos que la variable de interés X sigue una distribución normal de media μ y desviación típica σ y que el valor nominal μ_0 está centrado en el rango de tolerancias, pueden obtenerse (San Matías et al., 2004) sendas expresiones para las probabilidades $p_g(\delta)$ y $p_y(\delta)$ (véase la figura 2.9):

$$p_g(\delta) = \Phi\left(\delta + \frac{6C_p}{\lambda}\right) - \Phi\left(\delta - \frac{6C_p}{\lambda}\right), \quad (2.21)$$

$$p_y(\delta) = \Phi(\delta + 3C_p) - \Phi(\delta - 3C_p) - p_g(\delta), \quad (2.22)$$

donde Φ representa la función de distribución acumulada de una variable aleatoria normal tipificada, es decir:

$$\Phi(z) = \Pr(Z \leq z), \quad Z \sim N(0,1) \quad (2.23)$$

y

$$\lambda = \frac{USL - LSL}{W_G/2}, \quad \lambda \geq 2. \quad (2.24)$$

El parámetro λ representa la razón entre la anchura de la zona de especificaciones y la mitad de la anchura de la zona verde. Cabe advertir que al utilizar esta notación se está dando por supuesto que la anchura de la zona verde W_G no puede ser nula (aunque tiende a cero según aumenta

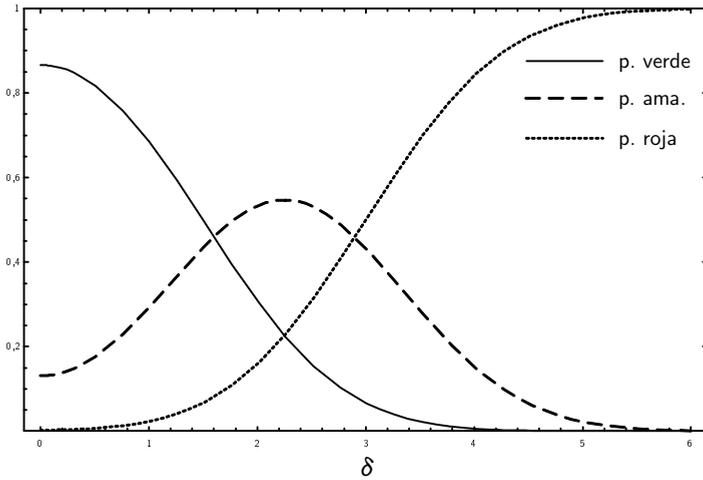


Figura 2.9: Probabilidades de obtener una pieza verde, amarilla y roja, en función de la desviación estandarizada δ , para un proceso con $C_p = 1$, en el caso del precontrol clásico ($\lambda = 4$).

el valor de λ).

Nótese que

$$0 < p_g(\delta) < 1, \quad (2.25)$$

ya que $p_g(\delta)$ representa la probabilidad de que una observación esté dentro de la zona verde $[G_L, G_U]$ (y ésta no se considera que pueda ser de anchura nula, según se acaba de remarcar), y que

$$0 \leq p_y(\delta) < 1, \quad (2.26)$$

puesto que la zona amarilla $[LSL, USL] \setminus [G_L, G_U]$ puede llegar a ser vacía; esto sucederá si la anchura de la zona verde W_G coincide con $USL - LSL$ (es decir, según (2.24), cuando $\lambda = 2$).

Por último, nótese también que en este enfoque es necesario conocer o tener una estimación de σ , o, lo que es lo mismo, una estimación de la capacidad potencial del proceso. Por ello, asumiremos conocida una estimación \hat{C}_p de dicha capacidad.

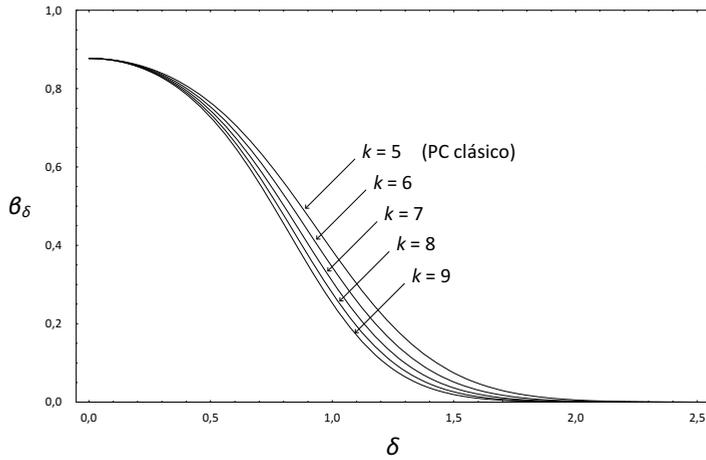


Figura 2.10: Curvas características del precontrol como contraste sobre δ , con $t = 2$ y $\alpha = 12,29\%$ (tasa de falsas alarmas en el precontrol clásico), para un proceso con $C_p = 1$. Elaboración propia a partir de una figura de [San Matías et al. \(2004\)](#).

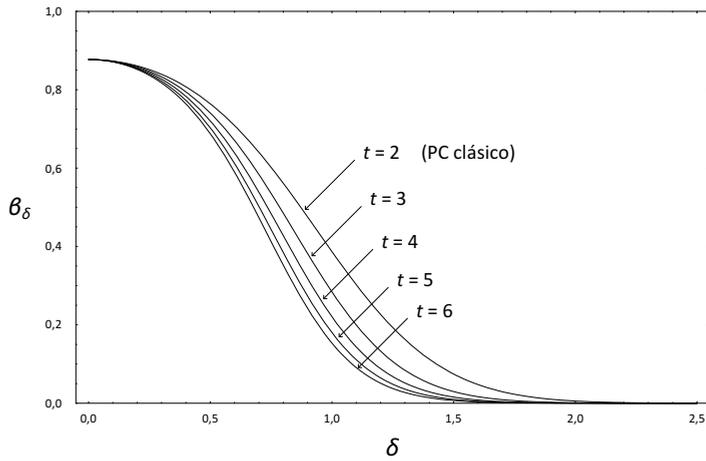


Figura 2.11: Curvas características del precontrol como contraste sobre δ , con $k = 5$ y $\alpha = 12,29\%$ (tasa de falsas alarmas en el precontrol clásico), para un proceso con $C_p = 1$. Elaboración propia a partir de una figura de [San Matías et al. \(2004\)](#).

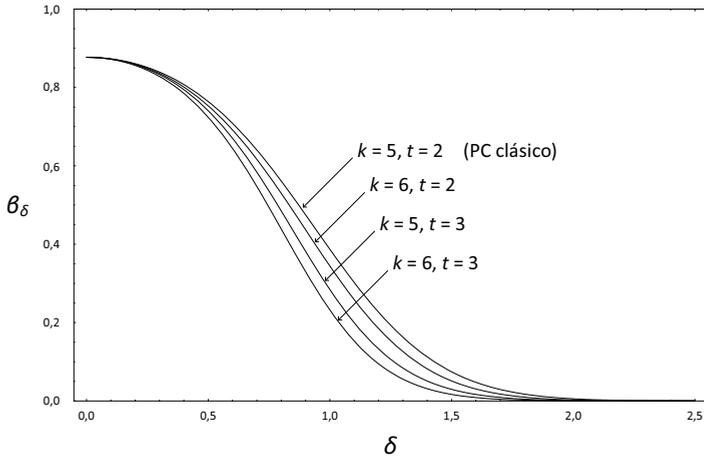


Figura 2.12: Curvas características del precontrol como contraste sobre δ , para diferentes valores de los parámetros k y t , y $\alpha = 12,29\%$ (tasa de falsas alarmas en el precontrol clásico), para un proceso con $C_p = 1$. Elaboración propia a partir de una figura de [San Matías et al. \(2004\)](#).

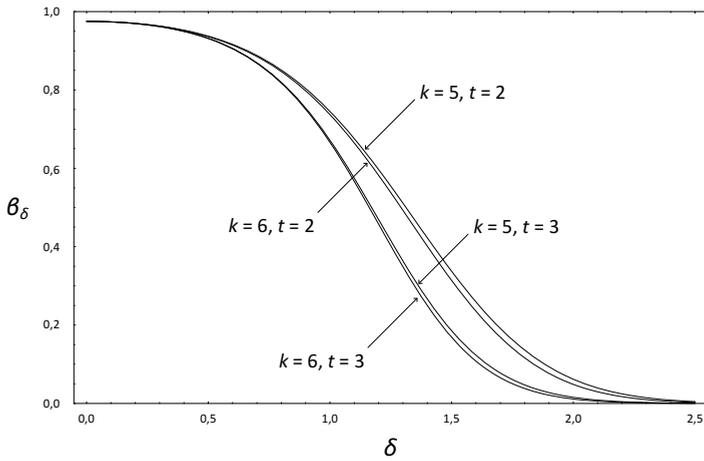


Figura 2.13: Curvas características del precontrol como contraste sobre δ , para diferentes valores de los parámetros k y t , y $\alpha = 2,5\%$, para un proceso con $C_p = 1$. Elaboración propia a partir de una figura de [San Matías et al. \(2004\)](#).

La curva característica del contraste, que permite evaluar la potencia del mismo, vendrá dada en este caso como la probabilidad de no rechazar H_0 en función de una desviación dada δ , y puede obtenerse a partir de (2.12).

Denotaremos por β_δ dicha probabilidad de aceptar H_0 cuando el proceso presenta una desviación δ (lo que se conoce habitualmente como error tipo II), es decir:

$$\beta_\delta = \Pr(\text{cualificar} \mid \delta). \quad (2.27)$$

Por tanto, dados diferentes planes de precontrol (es decir, dadas diferentes combinaciones para k , t y λ), y supuesta una estimación de la capacidad potencial del proceso \hat{C}_p , sus comportamientos pueden ser valorados y comparados entre sí a través de sus respectivas curvas características.

San Matías et al. señalan que, con un incremento de la anchura de la zona verde W_G en relación a la técnica clásica —lo cual equivale a disminuir el valor de λ —, se consigue reducir la tasa de falsa alarma α . Asimismo, muestran que un aumento de los parámetros k y t (sobre todo de éste último) mejora la potencia de la técnica. Esto se ilustra a través de las figuras 2.10, 2.11, 2.12 y 2.13. En el capítulo 4 del presente trabajo se demostrarán estas y otras propiedades.

En consecuencia, el precontrol generalizado, mediante una buena elección de los parámetros k , t y λ , consigue superar los principales problemas que presenta el precontrol clásico, que son la alta proporción de falsas alarmas y la baja potencia para detectar variaciones reales de la media (Mackertich, 1990), como ya hemos mencionado anteriormente. Es decir, *podemos asegurar la utilidad del precontrol generalizado como método para contrastar la estabilidad (de la media) del proceso.*

2.5.2. Precontrol como contraste sobre la capacidad

En el precontrol clásico, habitualmente se asume que la capacidad potencial del proceso es conocida y toma el valor mínimo aceptable,

$C_p = 1$, o valores mayores, como $C_p = 1,33$. Es en ese caso cuando el enfoque anterior, del precontrol visto como contraste sobre la desviación del proceso δ , es aplicable.

Dado que, en realidad, el objetivo principal del precontrol no es tanto detectar desviaciones en la media del proceso como evaluar la aptitud del mismo para generar piezas dentro de especificaciones, tiene sentido considerarlo como un modo de resolver un contraste de hipótesis sobre la capacidad potencial del proceso, en el caso en que ésta sea desconocida. Tomando 1 como valor mínimo aceptable para C_p , nos estamos refiriendo a esto:

$$\begin{cases} H_0 : C_p \geq 1 \\ H_1 : C_p < 1 \end{cases} \quad (2.28)$$

Cuando el resultado del precontrol sea cualificar el proceso (obtención de k verdes consecutivas), asumiremos como cierta H_0 . En caso contrario, rechazaremos la hipótesis, para un nivel de significación α , y por tanto podremos afirmar (con una probabilidad de error α) que el proceso no es capaz.

Estamos proponiendo aquí, por tanto, un enfoque alternativo y complementario al del precontrol como contraste sobre la media del proceso, tratado por [San Matías et al.](#) y desarrollado en la subsección anterior.

Para realizar este tipo de test, necesitaremos conocer o estimar μ o, equivalentemente, δ' , la razón entre la desviación absoluta de la media del proceso y la anchura de la zona de especificaciones:

$$\delta' = \frac{|\mu_0 - \mu|}{USL - LSL} \quad (2.29)$$

Nótese que, a partir de (2.11) y (2.29),

$$\delta = 6C_p\delta' \quad (2.30)$$

La probabilidad de cualificar el proceso sigue determinada por la expresión (2.12), aunque ahora las probabilidades de obtención de verde

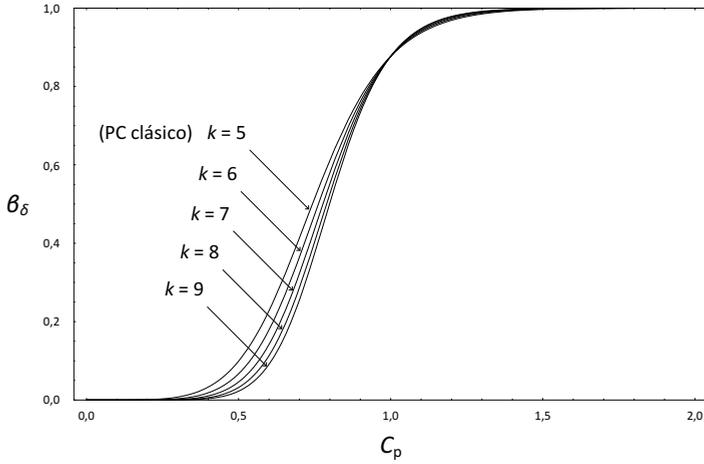


Figura 2.14: Curvas características del precontrol como contraste sobre C_p , con $t = 2$ y $\alpha = 12,29\%$ (tasa de falsas alarmas en el precontrol clásico), para un proceso con $\delta' = 0$.

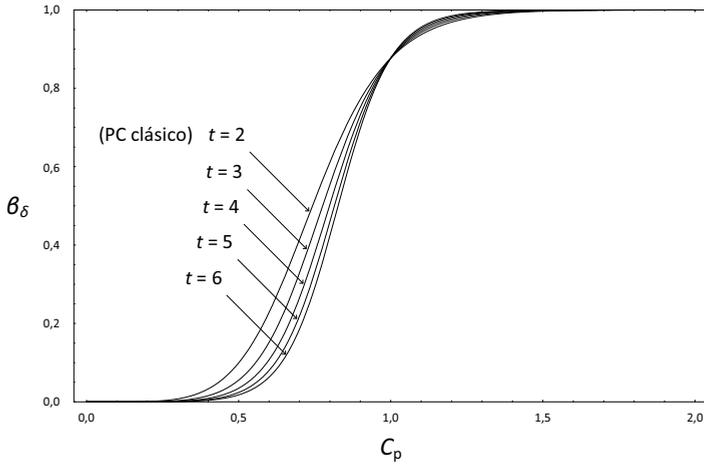


Figura 2.15: Curvas características del precontrol como contraste sobre C_p , con $k = 5$ y $\alpha = 12,29\%$ (tasa de falsas alarmas en el precontrol clásico), para un proceso con $\delta' = 0$.

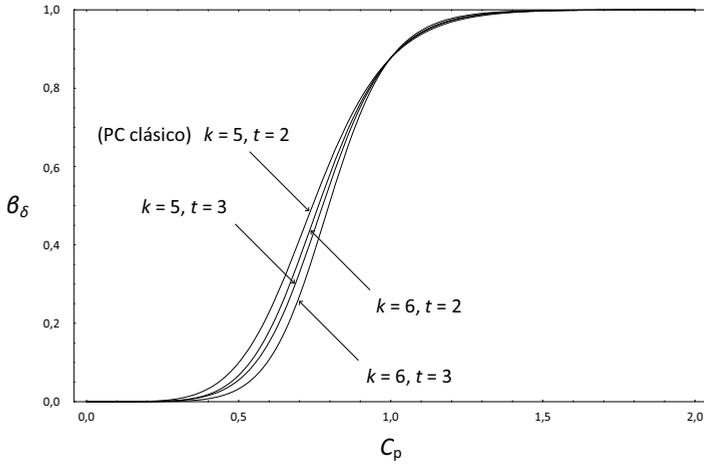


Figura 2.16: Curvas características del precontrol como contraste sobre C_p , para diferentes valores de los parámetros k y t , y $\alpha = 12,29\%$ (tasa de falsas alarmas en el precontrol clásico), para un proceso con $\delta' = 0$.

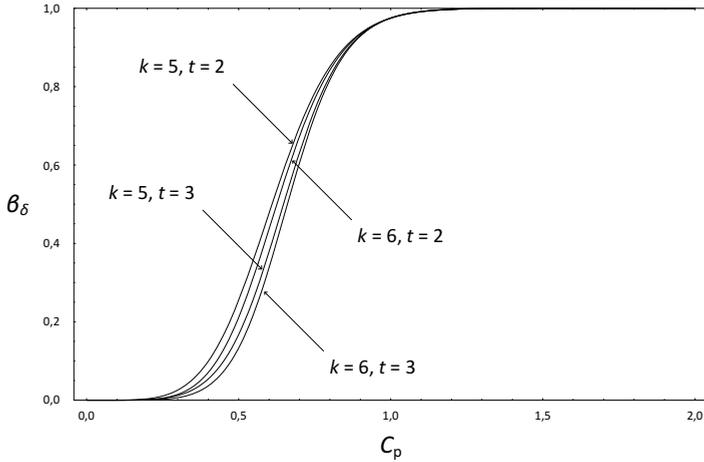


Figura 2.17: Curvas características del precontrol como contraste sobre C_p , para diferentes valores de los parámetros k y t , y $\alpha = 2,5\%$, para un proceso con $\delta' = 0$.

y de amarilla, p_g y p_y , serán función de la capacidad potencial C_p , y no de δ (ya que esta última se supone conocida, en este caso). Concretamente, si asumimos de nuevo normalidad de la característica de calidad X que estamos estudiando, se tiene:

$$p_g(C_p) = \Phi \left(6C_p \left(\frac{1}{\lambda} + \delta' \right) \right) - \Phi \left(-6C_p \left(\frac{1}{\lambda} - \delta' \right) \right), \quad (2.31)$$

$$p_y(C_p) = \Phi (3C_p (1 + 2\delta')) - \Phi (-3C_p (1 - 2\delta')) - p_g(C_p). \quad (2.32)$$

La tasa de falsa alarma para este contraste será:

$$\alpha = \sup_{C_p \geq 1} (1 - \Pr(\text{cualificar} \mid C_p)) = 1 - \Pr(\text{cualificar} \mid C_p = 1). \quad (2.33)$$

De nuevo, puede evaluarse el efecto de variar los parámetros del modelo, k , t y λ , por medio de la curva característica del contraste (véanse las figuras 2.14, 2.15, 2.16 y 2.17). También en este caso un aumento en los parámetros k y t (especialmente en el segundo) proporciona una mayor capacidad para detectar procesos no capaces. Asimismo, un incremento de la anchura de la zona verde favorece una disminución de α , la probabilidad de error tipo I.

De esta manera, hemos mostrado cómo el precontrol generalizado puede ser visto como un test sobre la media o sobre la capacidad del proceso, y que diferentes planes de precontrol pueden ser comparados en este contexto a través de sus correspondientes curvas características.

2.6. Un ejemplo práctico

Para completar este capítulo, ilustraremos el funcionamiento del precontrol generalizado a través de un ejemplo sencillo, extraído de [San Matías et al. \(2004\)](#).

Supongamos que la característica de calidad que estamos interesados en controlar es el diámetro interno de la cabeza de una biela (véase la

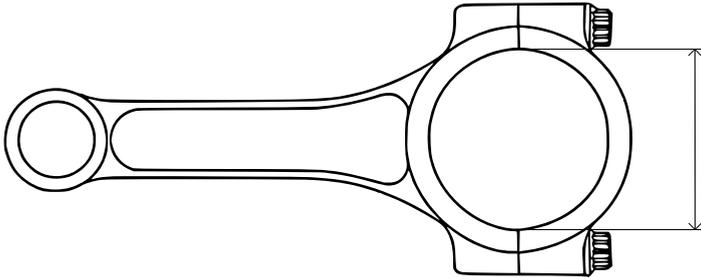


Figura 2.18: Croquis de una biela. Elaboración propia a partir de una imagen cedida gratuitamente por Lunati Co. (2008).

figura 2.18). El rango de tolerancias para dicha magnitud es $10 \pm 0,1$ cm. La capacidad potencial del proceso se supone constante y conocida, siendo de $C_p = 1,5$.

Queremos determinar un plan de precontrol (es decir, los valores de k , t y λ) de manera que verifique las siguientes condiciones:

- La tasa de falsas alarmas, α , debe ser inferior o igual al 1 %.
- La probabilidad de no detectar (y, por tanto, cualificar) un proceso que produzca una proporción de piezas fuera de especificaciones de $p^* = 2\%$ debe ser inferior o igual al 0,5 %.

En ocasiones, como sucede en este ejemplo, en lugar de hablarse de una desviación del proceso δ , se indica una proporción de piezas no conformes p , lo cual es, en realidad, más intuitivo y fácil de establecer y manejar por el usuario. Fijado y conocido el valor de C_p (y, por tanto, el valor de la desviación típica del proceso, σ), es posible, a partir de una proporción dada p^* de piezas no conformes, calcular la desviación δ^* que produciría dicha proporción, a través de la siguiente relación:

$$1 - p^* = \Phi(\delta^* + 3C_p) - \Phi(\delta^* - 3C_p), \quad (2.34)$$

que es válida siempre que el proceso siga una distribución normal y el valor nominal se encuentre en el centro del rango de tolerancias.

En este caso, si $p^* = 0,02$ y $C_p = 1,5$, por (2.34) se tiene que $\delta^* = 2,446$.

Por tanto, se desea un plan de precontrol que cumpla las siguientes dos condiciones:

$$1 - \Pr(\text{cualificar} \mid \delta = 0) \leq 0,01 \quad \text{y} \quad (2.35)$$

$$\Pr(\text{cualificar} \mid \delta = 2,446) \leq 0,005. \quad (2.36)$$

[San Matías et al.](#) muestran que es posible determinar diferentes combinaciones de los parámetros k , t y λ de manera que se verifiquen ambas restricciones. Concretamente, en la tabla 2.1 se muestra el valor de α y β_{δ^*} (los lados izquierdos de las restricciones (2.35) y (2.36), respectivamente), para distintos valores de los parámetros k y t , dejando fijo $\lambda = 4$ (su valor en el precontrol clásico). Por otro lado, en la tabla 2.2, fijado α a 0,01, su valor límite, y para diferentes valores de k y t , se ha calculado el valor de λ que hace que se verifique la inecuación (2.35) con igualdad; posteriormente se obtiene el valor de β_{δ^*} (el lado izquierdo de la segunda restricción) a partir de los parámetros ya fijados. En ambas tablas, se han destacado en negrita los casos en que se verifican simultáneamente las dos restricciones.

De entre todas las combinaciones encontradas que satisfacen las condiciones (2.35) y (2.36), podemos seleccionar dos de ellas, por el bajo valor requerido para k y t :

$$k = 5 \text{ y } t = 3, \text{ con } \lambda = 5,7204 \quad (2.37)$$

y

$$k = 6 \text{ y } t = 2, \text{ con } \lambda = 4,3576. \quad (2.38)$$

En resumen, de estos resultados se deduce que, en general, es posible encontrar diferentes combinaciones de los parámetros k , t y λ de modo que el plan de precontrol resultante verifique ciertas condiciones como las expresadas al inicio de este ejemplo. Asimismo, se ha mostrado que la liberación de dichos parámetros, propuesta por [San Matías et al.](#), permite mejorar el comportamiento de la técnica clásica.

k		t			
		2	3	4	5
5	α	0,3169 %	0,0113 %	0,0038 %	0,0037 %
	β_{δ^*}	3,4524 %	6,5086 %	10,4409 %	14,7973 %
6	α	0,3849 %	0,0138 %	0,0047 %	0,0045 %
	β_{δ^*}	1,4758 %	2,8337 %	4,6563 %	6,7819 %
7	α	0,4544 %	0,0163 %	0,0055 %	0,0053 %
	β_{δ^*}	0,6264 %	1,2124 %	2,0138 %	2,9707 %
8	α	0,5257 %	0,0188 %	0,0064 %	0,0061 %
	β_{δ^*}	0,2651 %	0,5148 %	0,8591 %	1,2744 %

Tabla 2.1: Valores de α y β_{δ^*} con $\lambda = 4$, para el ejemplo de la sección 2.6. En negrita se destaca el caso que hace que se verifiquen las dos restricciones. Elaboración propia a partir de la tabla presentada por [San Matías et al. \(2004\)](#).

k		t			
		2	3	4	5
5	β_{δ^*}	1,050937 %	0,092693 %	0,016782 %	0,004790 %
	λ	4,4444	5,7204	6,8448	7,8667
6	β_{δ^*}	0,465742 %	0,024616 %	0,003126 %	0,000689 %
	λ	4,3576	5,5884	6,6597	7,6228
7	β_{δ^*}	0,212152 %	0,006810 %	0,000613 %	0,000105 %
	λ	4,2874	5,4813	6,5089	7,4237
8	β_{δ^*}	0,098964 %	0,001951 %	0,000126 %	0,000017 %
	λ	4,2289	5,3916	6,3820	7,2560

Tabla 2.2: Valores de β_{δ^*} y λ con $\alpha = 0,01$, para el ejemplo de la sección 2.6. En negrita se destacan los casos que hacen que se verifique la restricción sobre β_{δ^*} . Elaboración propia a partir de la tabla presentada por [San Matías et al. \(2004\)](#).

2.7. Conclusiones

En este capítulo hemos dado una visión general del precontrol, como parte de las herramientas de control estadístico de procesos, y, en tal sentido, hemos realizado una breve comparación de esta técnica con

los gráficos de control. Se ha descrito la versión clásica y las diferentes mejoras propuestas en la literatura, haciendo especial hincapié en el precontrol generalizado, de [San Matías et al. \(2004\)](#). Se ha mostrado cómo éste consigue mejorar el comportamiento del método clásico, en términos de su capacidad para detectar situaciones disconformes, mediante la liberación de ciertos parámetros que rigen su funcionamiento, lo cual se ha ilustrado a través de un ejemplo práctico.

Cabe destacar que, hasta donde sabemos, no existe en la literatura una revisión del precontrol que recoja y estudie todas las variantes y propuestas que aquí se han presentado.

Además, como aportaciones propias al precontrol, en este capítulo hemos sugerido una nueva visión del precontrol generalizado como contraste sobre la capacidad del proceso, y hemos demostrado que la expresión desarrollada por [San Matías et al.](#) para la probabilidad de cualificar un proceso está bien definida, incluso en los casos en que $k = 1$ o $t = 1$ (proposición 2.1).

En el capítulo siguiente, partiremos del enfoque propuesto por [San Matías et al.](#) para desarrollar una propuesta propia de mejora del precontrol basada en técnicas de optimización.

Precontrol óptimo

3.1. Introducción

En el capítulo anterior hemos descrito la versión clásica del precontrol, así como algunas modificaciones que se han realizado sobre la misma con el objeto de mejorar su comportamiento. Concretamente, se ha analizado en mayor profundidad la propuesta de [San Matías et al. \(2004\)](#), consistente en dotar de una mayor flexibilidad al procedimiento en su fase inicial, y se ha mostrado que una adecuada elección del valor de los parámetros k , t y λ permite una reducción de la tasa de falsa alarma y un incremento de la potencia para detectar situaciones no conformes.

El problema reside justamente en la elección del valor de los parámetros, ya que esta decisión continúa en manos del usuario. Tal y como hemos ilustrado con un ejemplo numérico al final del capítulo anterior, puede existir más de una manera de elegir los valores de k , t y λ que garantice unos resultados aceptables, fijados los valores máximos para la tasa de falsa alarma y la probabilidad de no detectar un proceso con una cierta desviación crítica.

En el presente capítulo nuestra propuesta es abordar el problema de la determinación de los parámetros desde el punto de vista de la Optimización. Nuestro objetivo ha sido automatizar el proceso de elección del valor de los parámetros, de manera que sea *óptima* respecto a ciertos

criterios previamente fijados, requiera pocos datos de entrada y resulte transparente y fácil de aplicar para el usuario final. Para ello, hemos desarrollado una nueva metodología, que presentamos seguidamente.

En primer lugar, obtendremos una formulación del problema de optimización. Un aspecto importante del proceso de construcción del modelo, en el que nos detendremos, será la designación de un criterio en relación al cual optimizar la elección del valor los parámetros. Una vez escogido un criterio, desarrollaremos las expresiones de la función objetivo y las restricciones del modelo, para el caso habitual en que la variable de interés sigue una distribución normal.

A continuación, mostraremos que el problema generado se trata de un problema de programación no lineal entera mixta (a la cual nos referiremos normalmente por sus siglas en inglés, MINLP) y discutiremos brevemente la aplicación de algunas técnicas conocidas para la resolución de dicha clase de problemas.

Por último, mostraremos la utilidad del modelo construido, mediante su aplicación a una batería de ejemplos concretos.

3.2. Planteamiento de la determinación de los parámetros del precontrol¹ como problema de optimización

Abordamos el problema de la determinación de los parámetros del precontrol como problema de optimización. Nuestro objetivo es llegar a expresar el problema como modelo matemático. Para ello, recordemos que en todo modelo de optimización debemos identificar tres componentes fundamentales: las *variables*, la *función objetivo* y las *restricciones*.

¹Durante el presente capítulo se dará por supuesto que nos estamos refiriendo en todo momento al precontrol generalizado.

3.2.1. Variables del modelo

El objetivo del problema que nos ocupa es averiguar cuál es *la mejor decisión* a tomar sobre el valor de los parámetros del procedimiento de cualificación, en relación a un criterio todavía por determinar. Por tanto, las variables del modelo deben ser justamente los tres parámetros, k , t y λ .

Los parámetros k y t —número necesario de verdes y amarillas consecutivas para cualificar o no cualificar el proceso, respectivamente— son de naturaleza entera y, por su significado, han de tomar un valor mayor o igual a 1. El parámetro λ toma valores reales, y su mínimo valor es 2, dado que representa el número de veces que está contenida la mitad de la anchura de la zona verde, $W_G/2$, en la anchura de la zona de especificaciones, $USL - LSL$ (véase (2.24)).

Cabe decir que, aunque a partir de ahora nos referiremos a k , t y λ principalmente como las *variables* de nuestro modelo, otra expresión aplicable a ellas, bastante común en el contexto del *diseño óptimo* de herramientas de control de calidad (y de procesos o sistemas en general), sería la de *parámetros de diseño* (del precontrol, en nuestro caso).

3.2.2. Criterios de optimización

Un aspecto fundamental en el presente enfoque es la elección de un criterio en relación al cual optimizar la elección del valor de las variables. Existen diferentes alternativas razonables, como, por ejemplo:

- Minimizar el tamaño muestral esperado para finalizar el procedimiento de cualificación, n .
- Minimizar la tasa de falsas alarmas, α .
- Maximizar la potencia, $1 - \beta$, para una desviación inaceptable respecto de la situación de control del proceso.
- Maximizar la *average run length* (ARL, longitud media de racha)

cuando el proceso está bajo control, o minimizarla para una desviación inaceptable.

- Maximizar el *average number inspected* (ANI, número medio de piezas inspeccionadas) cuando el proceso está bajo control, o minimizarlo para una desviación inaceptable.
- Minimizar el coste estimado total del proceso.

Se trata de valores que son tenidos en cuenta habitualmente a la hora de medir el comportamiento de una herramienta de control de calidad. A continuación pasamos a describir con más detalle cada una de estas alternativas.

Tamaño muestral esperado

Dado un plan de precontrol concreto (es decir, una vez fijado el valor de los parámetros), llamaremos n al tamaño muestral total necesario para terminar el procedimiento de cualificación. Para un determinado proceso, éste finaliza en el momento en que el muestreo produzca k piezas verdes consecutivas, o bien t amarillas consecutivas, o bien una roja. Así pues, la cantidad n es una variable aleatoria discreta, cuya distribución depende de las características del proceso (concretadas en su desviación δ y en su capacidad C_p) y de los parámetros del precontrol k , t y λ .

Un posible criterio de optimización es el de buscar la combinación de parámetros (k, t, λ) que, para un proceso determinado, minimice el valor esperado de n ; es decir, obtener el plan de precontrol que, en término medio, necesite muestrear la menor cantidad posible de piezas para finalizar. Esto tiene especial interés, sobre todo, en procesos en que el muestreo pueda resultar costoso.

Más adelante obtendremos una expresión para $E[n]$.

Tasa de falsa alarma

Hemos definido la tasa de falsa alarma o probabilidad de error de tipo I, α , como la probabilidad de que un plan de precontrol dado pueda

rechazar (no cualificar) un proceso que en realidad es correcto, bien en el sentido de que está centrado (véase la expresión (2.20)) o bien en el sentido de que es un proceso capaz (2.33).

Así, buscar un plan de precontrol que minimice esta tasa de falsa alarma α supondría otra alternativa como criterio de optimización.

En cualquier caso, es necesario que dicha probabilidad de error tipo I sea tenida en cuenta de alguna manera en el modelo que elaboremos, si deseamos que éste proporcione una alternativa válida al precontrol clásico.

Potencia para detectar procesos no conformes

Sea β la probabilidad de que el precontrol termine cualificando un proceso no conforme, bien porque está descentrado, bien porque no es capaz. Es decir, β es la probabilidad de error tipo II y, por tanto, $1 - \beta$, la probabilidad de no cualificar un proceso cuando no es correcto, es lo que se conoce como *potencia* de la técnica, en cualquiera de los dos enfoques expuestos en la sección 2.5.

El tercer criterio que proponemos para optimizar la elección del valor de los parámetros es maximizar la potencia del precontrol para detectar una cierta desviación del proceso o una cierta capacidad potencial inaceptables.

Si no se tomase como valor a optimizar, resulta igualmente obvio que la probabilidad de error tipo II debería estar limitada o controlada en el modelo que construyamos.

Más adelante detallaremos las expresiones para α y β en el caso del precontrol generalizado, obtenidas por [San Matías et al. \(2004\)](#).

Longitud media de racha (ARL)

La *longitud media de racha* (ARL, por sus siglas en inglés) es un parámetro usado con frecuencia para valorar el comportamiento de herramientas para el control estadístico de procesos, tales como gráficos de control. Se define como la esperanza matemática del número de

(sub)muestras que es necesario extraer para obtener una señal de falta de control.

Hay que tener en cuenta que, tal y como se reflejó en la subsección 1.2.2, cuando se habla de *muestra* en el contexto de gráficos de control es para referirse a cada uno de los subconjuntos de observaciones (de tamaño constante o no) que se utilizan para generar un punto del gráfico. Por tanto, *no* coincide con el concepto de muestra que hemos utilizado anteriormente al definir el parámetro *tamaño muestral* del precontrol, que aquí hace referencia al número total de piezas que es necesario inspeccionar para terminar el procedimiento. En el presente apartado nos referiremos como *submuestra* al conjunto de observaciones que dan lugar a un punto en un gráfico de control.

En el contexto de gráficos de control, la ARL cuantifica, de alguna forma, la *rapidez* con que se detecta que un proceso está descentrado cuando está operando fuera de control. Es, por tanto, una medida de la *eficiencia* del procedimiento.

Trasladar esta idea al caso del precontrol *no* es del todo trivial. Por la manera en como están diseñados los gráficos de control y por el objetivo que persiguen, en su ejecución los muestreos se suceden indefinidamente hasta que se produce una señal de falta de control. En cambio, en el precontrol el objetivo es realizar una valoración inicial del proceso (y no una monitorización constante del mismo) y por ello el muestreo de piezas se detiene cuando se obtienen evidencias suficientes tanto para cualificar el proceso (k piezas verdes consecutivas) como para no cualificarlo (t amarillas consecutivas o 1 roja). Esto ya sugiere, por sí mismo, que un concepto como el ARL no sea quizás el más adecuado para medir la eficiencia del precontrol.

Para poder llegar a obtener una expresión para la ARL en el precontrol, tendríamos que preguntarnos cuál es la equivalencia correcta en esta técnica del concepto de *(sub)muestra* en gráficos de control. Por un lado, en el precontrol cada pieza inspeccionada genera un resultado (verde, amarillo o rojo) que es graficado o simplemente registrado; por tanto, en paralelismo con los gráficos de control, podríamos decir que cada

submuestra es de tamaño unitario, y por tanto el número esperado de submuestras hasta obtener una señal de alarma (definición de la ARL) coincidiría con el número esperado de piezas a extraer hasta dicha señal. El problema de este enfoque es que, a diferencia de los gráficos de control, el precontrol puede finalizar su ejecución cualificando el proceso y sin mostrar, por tanto, ninguna señal de alerta.

Existe también otra manera de verlo: tal y como se ha discutido en la subsección 1.2.2, puede considerarse que en cada muestreo de un gráfico de control (es decir, en cada punto del gráfico) se está realizando un contraste de hipótesis sobre el parámetro objeto de control (por ejemplo, la media real del proceso μ), donde la hipótesis nula es que el proceso está centrado. Desde ese punto de vista, lo que mide la ARL es cuántas veces, en promedio, es necesario realizar ese contraste para que termine rechazando la hipótesis nula del mismo (por ejemplo, $\mu = \mu_0$); de ahí que la expresión para la ARL de un gráfico de control de Shewhart para un proceso que opera descentrado como consecuencia de una causa asignable se calcule tradicionalmente como la esperanza matemática de una distribución geométrica con probabilidad de éxito $p = 1 - \beta$ (la probabilidad de que una realización del contraste rechace que el proceso está centrado), es decir, $ARL = 1/p = 1/(1 - \beta)$. De forma similar, la ARL de un gráfico de control para un proceso centrado se calcula como $1/\alpha$, siendo α la tasa de falsa alarma.

Si intentamos traducir esto al caso del precontrol, hemos de tener en cuenta que en el precontrol *todo el gráfico* (es decir, el total de piezas muestreadas hasta finalizar el procedimiento) actúa como *un único* contraste de hipótesis, que finaliza rechazando o no que el proceso está centrado (véase la subsección 2.5.1). Así, calcular la ARL en el caso del precontrol, de manera que sea equiparable al modo en que se calcula en gráficos de control, equivaldría a contabilizar cuántas veces hay que repetir el procedimiento del precontrol para terminar no cualificando el proceso; sería como si cada ejecución del procedimiento de precontrol equivaliese a *un punto* de un hipotético gráfico de control. Con este enfoque, la expresión para la ARL del precontrol sería la misma que

en los gráficos de control tradicionales, es decir, $ARL = 1 / (1 - \beta)$ para procesos descentrados y $ARL = 1/\alpha$ cuando el proceso esté centrado.

Es esta última interpretación (es decir, entender que cada ejecución del procedimiento de precontrol equivale a *una submuestra* a efectos de calcular la ARL) la que consideramos más acertada, ya que de esta forma la equiparamos a la definición de la ARL en gráficos de control, lo cual permitiría realizar hipotéticas comparaciones entre el precontrol y cualquier herramienta de control de procesos que utilice la ARL como medida de eficiencia.

Sin embargo, este enfoque tiene el inconveniente de no ajustarse a la manera en que se implementa el precontrol (en particular, la fase de cualificación, que es la que estamos tratando aquí): normalmente, dicha técnica *no* se ejecuta iterativamente sobre un proceso, sino que se aplica una sola vez sobre el mismo para terminar cualificándolo o no. Así, la ARL, definida de esa forma, quedaría como un parámetro teórico y poco conectado con el modo real de llevar a cabo el precontrol.

En conclusión, como ya se ha dicho, la ARL no es el parámetro más indicado para medir el comportamiento del precontrol. En el problema de optimización que construiremos, no tendremos directamente en cuenta la ARL, pero sí las probabilidades de error α y β ya mencionadas y que, como se acaba de mostrar, tienen una relación directa con la ARL.

Profundizar en una definición adecuada para la ARL en el caso del precontrol es una tarea sin duda interesante y que, no obstante, queda fuera de los objetivos del presente trabajo.

Número medio de unidades inspeccionadas (ANI)

El parámetro *número medio de unidades inspeccionadas* (*average number inspected*, ANI; Bourke, 1991) se define como la esperanza matemática del número de unidades a inspeccionar u observaciones a realizar hasta obtener una señal de falta de control. Surge como alternativa a la ARL, cuando el objetivo es evaluar o comparar herramientas de control de calidad con tamaños de submuestras distintos entre sí; en tales situaciones, es más representativo utilizar como unidad de medida el número de

piezas inspeccionadas que el número de submuestras extraídas. Es también, por tanto, una medida de eficiencia.

Las dos formas de trasladar el concepto de ARL al caso del precontrol que hemos presentado en el apartado anterior darían aquí lugar a sendas posibles maneras de definir el ANI.

Si consideramos *cada observación* del precontrol como una submuestra de tamaño unitario, entonces el ARL coincidiría con el ANI, y ambos conceptos serían parecidos, aunque no iguales, al de tamaño muestral esperado $E[n]$: la diferencia es que la ARL y el ANI miden *cuánto hay que esperar* en término medio hasta que aparezca una observación roja o t amarillas consecutivas, mientras que el tamaño muestral que manejamos nosotros mide la duración del precontrol hasta que finalice, bien cualificando el proceso o bien no cualificándolo. Aquí nos encontramos con el mismo problema de definición que descubríamos en el caso de la ARL: ¿cómo definir la longitud de racha de una ejecución en la que no llegue a producirse una alerta porque se observen antes k verdes consecutivas?

Si, en cambio, consideramos *cada ejecución* del precontrol como una submuestra, entonces claramente podríamos definir el ANI como $E[n]$ ARL. Este enfoque presenta los mismos inconvenientes que la correspondiente definición de ARL, los cuales ya han sido expuestos.

En resumen, al igual que no se ha considerado la ARL como la medida más adecuada para valorar el comportamiento del precontrol, el ANI no será incluido directamente en el modelo de optimización que construyamos, aunque sí estará presente un concepto muy similar: el tamaño muestral esperado $E[n]$, que consideramos más acertado como medida de la eficiencia de la técnica.

Coste total del proceso

En general, muchos otros enfoques pueden ser propuestos, pero quizás, desde un punto de vista práctico y económico, sería muy interesante buscar aquella combinación de parámetros que minimice el coste total esperado del procedimiento de cualificación. Es un enfoque que, en gráficos de control, ha sido objeto de investigación en tiempos

relativamente recientes (para más información véase, por ejemplo, el capítulo 10 de [Montgomery, 2012](#)).

Esta propuesta supondría poder estimar de manera fiable cantidades como, entre otras, las siguientes:

- El coste que supone muestrear cada unidad; habitualmente éste consta de un término fijo y uno variable por unidad muestreada.
- El coste de localizar y reparar la causa de un proceso no conforme.
- El coste por detener y revisar un proceso que realmente no lo necesita, también referido como coste de investigar una falsa alarma (coste asociado a un error tipo I).
- El coste asociado a producir y distribuir unidades en un proceso no conforme, en función de la desviación o de la proporción de piezas no conformes (coste asociado a un error tipo II).

A partir de dichas estimaciones y de las expresiones para $E[n]$, α y β , podría construirse una función objetivo que representase adecuadamente el coste total esperado del procedimiento de precontrol.

Este enfoque es, seguramente, el más completo, si bien al mismo tiempo supone una mayor complejidad para el usuario final². Abordarlo en toda su profundidad requiere un análisis más completo y exhaustivo de los costes asociados, lo cual excede los límites propuestos para esta tesis, si bien es uno de nuestros objetivos de trabajo futuro.

Elección de un criterio de optimización

En el presente trabajo hemos elegido como criterio de optimización el de minimizar el valor esperado del tamaño muestral n . Como hemos mencionado, este enfoque puede resultar especialmente interesante

²Al hablar de *usuario final* nos referimos en este caso no tanto al operario que está a pie de línea aplicando el precontrol, como al empleado encargado de realizar su diseño, quien debería, entre otras cosas, disponer de información para conocer o al menos estimar todos los costes que se han mencionado.

cuando el muestreo sea costoso o, en general, siempre que se desee que el precontrol se realice en el menor tiempo y con el menor esfuerzo posible.

Como se ha avanzado, el control acerca de la tasa de falsa alarma α y de la potencia para detectar ciertas desviaciones $1 - \beta$ se realizará a través de sendas restricciones.

3.2.3. Restricciones a contemplar

En la subsección 3.2.2 se han enumerado diferentes parámetros relacionados con el comportamiento del precontrol (comunes a otras herramientas de control de calidad) que suele ser de interés controlar en la práctica. Una vez elegido como valor a optimizar el tamaño muestral, y tras todo lo dicho en esa subsección, es razonable y adecuado que el modelo de optimización que se va a construir contemple como restricciones las siguientes dos limitaciones:

- Una cota superior para la tasa de falsa alarma o probabilidad de error de tipo I α del plan de precontrol generado.
- Una cota superior para la probabilidad β de que el precontrol no detecte un cierta desviación inaceptable del proceso (error de tipo II).

Los dos conceptos han sido ya introducidos en la subsección referida. Cabe añadir que, a nivel práctico, tener controlada la tasa de falsa alarma (primera restricción) permite limitar la frecuencia con que el precontrol determina que es necesario detener y revisar un proceso cuando realmente no lo es, mientras que el requisito sobre la probabilidad de no detectar un proceso no conforme (segunda restricción) reduce el riesgo de producir un bien o servicio que sea rechazado por el cliente. Desde el punto de vista del control de la calidad, es fundamental contar con herramientas que permitan vigilar ambos aspectos simultáneamente.

3.3. Desarrollo del modelo de optimización para el precontrol generalizado

A continuación, vamos a desarrollar el modelo de optimización para el precontrol generalizado, según los criterios expuestos en la sección anterior. Vamos a considerar el caso en que la característica de calidad siga una distribución normal. Tal y como ya se ha dicho con anterioridad, veremos que se trata de un problema de MINLP.

3.3.1. El modelo general

Según la decisión tomada sobre el criterio de optimización, la función objetivo a minimizar será el valor esperado del tamaño muestral, es decir, $E[n]$.

Además, incluiremos en nuestro planteamiento los requisitos que deberá verificar el plan de precontrol óptimo ya mencionados: sendas cotas superiores para las probabilidades de error tipo I y tipo II del procedimiento.

Así, el plan de precontrol que buscamos deberá ser el que minimice el tamaño muestral esperado, de manera que, al mismo tiempo, la tasa de falsa alarma no sea superior a una cierta cota superior α^* y tal que la probabilidad β (para un δ^* o un C_p^* dados) de dar como bueno un proceso no correcto no supere un β^* dado.

Por tanto, de manera esquemática, el modelo que responde a nuestro planteamiento es el siguiente:

$$\left. \begin{array}{l} \text{Min} \quad E[n] \\ \text{s.a:} \quad \alpha \leq \alpha^* \\ \quad \quad \beta \leq \beta^* \\ \quad \quad k, t \geq 1 \\ \quad \quad \lambda \geq 2 \\ \quad \quad k, t \in \mathbb{Z}, \lambda \in \mathbb{R} \end{array} \right\} \quad (3.1)$$

3.3.2. Hipótesis y notación

Como acabamos de decir, desarrollaremos el modelo (3.1) para el caso en que se esté interesado en controlar una característica de calidad continua unidimensional X que sigue una distribución normal de media μ y desviación típica σ , y cuyo valor nominal μ_0 se encuentra en el punto medio del rango de tolerancias $[LSL, USL]$.

Existen dos opciones: si se pretende usar el precontrol como un test sobre la desviación δ (véase la subsección 2.5.1), supondremos conocida la capacidad potencial del proceso, C_p (1.1). Alternativamente, en el caso de que estuviéramos considerando el precontrol como un contraste sobre la capacidad potencial del proceso (véase la subsección 2.5.2), tomaríamos como conocida la razón δ' (2.29).

Vamos a desarrollar el modelo para el caso en que supongamos conocida C_p o una estimación suya, lo que equivale a decir que podemos conocer o estimar σ , la variabilidad real del proceso. En este enfoque, lo que permanece desconocido es μ , la media real del proceso (y es por eso precisamente que el precontrol puede ser considerado en este caso como un contraste sobre la media, como ya se ha visto en la subsección 2.5.1).

Notemos que el tamaño muestral n es una variable aleatoria cuya distribución (y, por tanto, su esperanza) dependerá de la desviación estandarizada del proceso, δ . Por ello, en general, nos referiremos a dicha cantidad como $n(\delta)$; en algunas ocasiones seguiremos utilizando simplemente n , por sencillez de notación, aun manteniendo presente que su distribución depende de δ .

Por otro lado, definiremos los siguientes *estados* por los que puede pasar el proceso cuando se está llevando a cabo el procedimiento de cualificación:

- Estado 0 o *estado inicial*, previo a muestrear la primera pieza.
- Estado F o *estado final*, inmediatamente posterior a la última extracción. Se puede llegar a esta situación bien tras haber obtenido k piezas verdes consecutivas, bien tras t amarillas consecutivas, o bien tras 1 roja.

- Estado iG , que se produce cuando la última pieza extraída ha sido verde y con ella se han acumulado i extracciones consecutivas de piezas verdes ($i = 1, \dots, k - 1$).
- Estado jY , que se produce cuando la última pieza extraída ha sido amarilla y con ella se han acumulado j piezas amarillas consecutivas ($j = 1, \dots, t - 1$).

Nótese que en todo momento el sistema se encuentra en uno de estos $k + t$ estados distintos.

En un proceso con desviación estandarizada δ , denotaremos por $n_{a,b}(\delta)$ al número de piezas que se deben muestrear hasta alcanzar el estado b partiendo del estado a . En particular, $n_{a,F}(\delta)$ será el número de extracciones necesario para llegar al estado final cuando el proceso se encuentra en el estado a . Entonces, llamaremos x_a^δ a la esperanza matemática de dicha variable aleatoria:

$$x_a^\delta = E[n_{a,F}(\delta)]. \quad (3.2)$$

3.3.3. Obtención de la función objetivo

Vamos a obtener una expresión analítica para la función objetivo del problema (3.1), $E[n(\delta)]$, el valor esperado de la cantidad de piezas a muestrear para finalizar el proceso de cualificación. Según la notación introducida en la subsección 3.3.2, $n(\delta) = n_{0,F}(\delta)$, y por tanto, $E[n(\delta)] = x_0^\delta$, es decir, el número esperado de extracciones necesarias para llegar al estado final a partir del estado inicial.

La variable aleatoria discreta $n_{0,F}(\delta)$ se rige por la siguiente función de probabilidad:

$$n_{0,F}(\delta) = \begin{cases} 1 + n_{1G,F}(\delta) & \text{con probabilidad } p_g(\delta) \\ 1 + n_{1Y,F}(\delta) & \text{con probabilidad } p_y(\delta) \\ 1 & \text{con probabilidad } p_r(\delta) \end{cases}, \quad (3.3)$$

donde $p_g(\delta)$ y $p_y(\delta)$ representan la probabilidad de extraer una pieza ver-

de y amarilla, respectivamente (véanse (2.21) y (2.22)), y $p_r(\delta)$ representa la probabilidad de muestrear una pieza roja, es decir:

$$p_r(\delta) = 1 - p_g(\delta) - p_y(\delta). \quad (3.4)$$

Nótese que

$$0 < p_r(\delta) < 1, \quad (3.5)$$

ya que $p_r(\delta)$ representa la probabilidad de que una medición esté fuera del intervalo $[LSL, USL]$, con $LSL < USL$ según (1.2).

Aplicando esperanzas en la expresión (3.3) obtenemos:

$$x_0^\delta = (1 + x_{1G}^\delta)p_g(\delta) + (1 + x_{1Y}^\delta)p_y(\delta) + p_r(\delta), \quad (3.6)$$

lo cual, teniendo en cuenta (3.4), puede expresarse de la siguiente forma:

$$x_0^\delta = 1 + p_g(\delta)x_{1G}^\delta + p_y(\delta)x_{1Y}^\delta. \quad (3.7)$$

En general:

$$n_{iG,F}(\delta) = \begin{cases} 1 + n_{(i+1)G,F}(\delta) & \text{con probabilidad } p_g(\delta) \\ 1 + n_{1Y,F}(\delta) & \text{con probabilidad } p_y(\delta) \\ 1 & \text{con probabilidad } p_r(\delta) \end{cases}, \quad (3.8)$$

para $i = 1, \dots, k - 2$, y

$$n_{(k-1)G,F}(\delta) = \begin{cases} 1 + n_{1Y,F}(\delta) & \text{con probabilidad } p_y(\delta) \\ 1 & \text{con probabilidad } p_g(\delta) + p_r(\delta) \end{cases}. \quad (3.9)$$

Análogamente, para los estados jY , $j = 1, \dots, t - 1$:

$$n_{jY,F}(\delta) = \begin{cases} 1 + n_{1G,F}(\delta) & \text{con probabilidad } p_g(\delta) \\ 1 + n_{(j+1)Y,F}(\delta) & \text{con probabilidad } p_y(\delta) \\ 1 & \text{con probabilidad } p_r(\delta) \end{cases}, \quad (3.10)$$

para $j = 1, \dots, t - 2$, y

$$n_{(t-1)YF}(\delta) = \begin{cases} 1 + n_{1G,F}(\delta) & \text{con probabilidad } p_g(\delta) \\ 1 & \text{con probabilidad } p_y(\delta) + p_r(\delta) \end{cases}. \quad (3.11)$$

Procediendo de manera análoga al caso de $n_{0,F}(\delta)$, aplicamos esperanzas en las expresiones (3.8), (3.9), (3.10) y (3.11) y, teniendo en cuenta de nuevo (3.4), llegamos a las expresiones siguientes:

$$x_{iG}^\delta = 1 + p_g(\delta)x_{(i+1)G}^\delta + p_y(\delta)x_{1Y}^\delta, \quad \text{para } i = 1, \dots, k - 2, \quad (3.12)$$

$$x_{(k-1)G}^\delta = 1 + p_y(\delta)x_{1Y}^\delta, \quad (3.13)$$

$$x_{jY}^\delta = 1 + p_g(\delta)x_{1G}^\delta + p_y(\delta)x_{(j+1)Y}^\delta, \quad \text{para } j = 1, \dots, t - 2, \quad (3.14)$$

$$x_{(t-1)Y}^\delta = 1 + p_g(\delta)x_{1G}^\delta. \quad (3.15)$$

Uniendo (3.7), (3.12), (3.13), (3.14) y (3.15) se obtiene un sistema con tantas ecuaciones como incógnitas ($k + t - 1$):

$$\left. \begin{aligned} x_0^\delta &= 1 + p_g(\delta)x_{1G}^\delta + p_y(\delta)x_{1Y}^\delta \\ x_{1G}^\delta &= 1 + p_g(\delta)x_{2G}^\delta + p_y(\delta)x_{1Y}^\delta \\ &\dots \\ x_{(k-2)G}^\delta &= 1 + p_g(\delta)x_{(k-1)G}^\delta + p_y(\delta)x_{1Y}^\delta \\ x_{(k-1)G}^\delta &= 1 + p_y(\delta)x_{1Y}^\delta \\ x_{1Y}^\delta &= 1 + p_g(\delta)x_{1G}^\delta + p_y(\delta)x_{2Y}^\delta \\ &\dots \\ x_{(t-2)Y}^\delta &= 1 + p_g(\delta)x_{1G}^\delta + p_y(\delta)x_{(t-1)Y}^\delta \\ x_{(t-1)Y}^\delta &= 1 + p_g(\delta)x_{1G}^\delta \end{aligned} \right\}. \quad (3.16)$$

Resolviendo el sistema (3.16) obtenemos las siguientes expresiones

para x_{1G}^δ y x_{1Y}^δ :

$$x_{1G}^\delta = \frac{\sum_{i=0}^{k-2} p_g(\delta)^i \sum_{j=0}^{t-1} p_y(\delta)^j}{1 - p_g(\delta)p_y(\delta) \sum_{i=0}^{k-2} p_g(\delta)^i \sum_{j=0}^{t-2} p_y(\delta)^j}, \quad (3.17)$$

$$x_{1Y}^\delta = \left(1 + p_g(\delta)x_{1G}^\delta\right) \sum_{j=0}^{t-2} p_y(\delta)^j. \quad (3.18)$$

Finalmente, sustituyendo (3.17) y (3.18) en (3.7), conseguimos una expresión para la función objetivo, $E[n(\delta)]$:

$$\begin{aligned} E[n(\delta)] &= x_0^\delta \\ &= \sum_{j=0}^{t-1} p_y(\delta)^j \left(1 + \frac{p_g(\delta) \sum_{i=0}^{k-2} p_g(\delta)^i \sum_{j=0}^{t-1} p_y(\delta)^j}{1 - p_g(\delta) \sum_{i=0}^{k-2} p_g(\delta)^i p_y(\delta) \sum_{j=0}^{t-2} p_y(\delta)^j} \right) \\ &= \frac{\sum_{i=0}^{k-1} p_g(\delta)^i \sum_{j=0}^{t-1} p_y(\delta)^j}{1 - \sum_{i=1}^{k-1} p_g(\delta)^i \sum_{j=1}^{t-1} p_y(\delta)^j}. \end{aligned} \quad (3.19)$$

Resolviendo el sistema (3.16) para el caso particular en el que $k = 1$ o $t = 1$, obtenemos

$$x_0 = \sum_{i=0}^{k-1} p_g(\delta)^i \sum_{j=0}^{t-1} p_y(\delta)^j,$$

que puede reducirse a la expresión general (3.19) aplicando (2.14).

Podemos simplificar la expresión (3.19) utilizando que el resultado de la suma parcial de una serie geométrica de razón $r \neq 1$ es

$$\sum_{i=0}^{m-1} r^i = \frac{1 - r^m}{1 - r}, \quad (3.20)$$

con lo cual:

$$E[n(\delta)] = \frac{(1 - p_g(\delta)^k) (1 - p_y(\delta)^t)}{(1 - p_g(\delta)) (1 - p_y(\delta)) - (p_g(\delta) - p_g(\delta)^k) (p_y(\delta) - p_y(\delta)^t)}. \quad (3.21)$$

Discusión sobre el valor de δ en la función objetivo

La distribución de $n(\delta)$ —y, por tanto, su esperanza— depende directamente de cuál sea la desviación real del proceso, como ya hemos hecho notar y se ha puesto de manifiesto al desarrollar su expresión analítica. Nótese que δ no es un parámetro cuyo valor podamos decidir o elegir. En general, desconocemos qué valores puede tomar y con qué frecuencia. Esto significa que *no* podemos pretender minimizar en general el tamaño muestral esperado necesario para llevar a cabo el procedimiento del precontrol, sea cual sea la desviación real del proceso, sino que deberemos realizar alguna asunción acerca del valor de δ en la función objetivo.

La primera solución que surge, de manera natural, sería intentar minimizar el peor caso, es decir, el mayor valor esperado de $n(\delta)$ para cualquier posible δ , lo cual podría expresarse como

$$\text{Min } \max_{\delta} E[n(\delta)] \quad (3.22)$$

De manera equivalente, lo anterior puede expresarse así:

$$\begin{aligned} \text{Min } \epsilon \\ \text{s.a: } E[n(\delta)] \leq \epsilon \text{ para } \delta \in [0, +\infty[\end{aligned} \quad (3.23)$$

La restricción del modelo (3.23) debe cumplirse para todo valor de $\delta \geq 0$. Es decir, en realidad no tenemos una restricción, sino infinitas, una para cada valor distinto del parámetro δ .

Por tanto, este enfoque nos conduciría a plantearnos un problema de programación semiinfinita (Goberna y López, 1998), lo cual excede las pretensiones del presente trabajo. De cualquier forma, el estudio del peor

caso será retomado de nuevo en la subsección 3.5.3.

La alternativa más directa y clara es la de fijar el valor del parámetro δ en la función objetivo a un δ_0 dado. En particular, elegimos como δ_0 el valor 0; es decir, nuestro modelo permitirá encontrar la configuración de parámetros del precontrol k , t y λ que minimicen $E[n(0)]$, el número esperado de piezas a muestrear para finalizar el procedimiento de cualificación *cuando el proceso se encuentra centrado*.

Por tanto, las expresiones (3.19) y (3.21) para la función objetivo, quedan finalmente así:

$$E[n(0)] = \frac{\sum_{i=0}^{k-1} p_g(0)^i \sum_{j=0}^{t-1} p_y(0)^j}{1 - \sum_{i=1}^{k-1} p_g(0)^i \sum_{j=1}^{t-1} p_y(0)^j} \quad (3.24)$$

$$= \frac{(1 - p_g(0)^k) (1 - p_y(0)^t)}{(1 - p_g(0)) (1 - p_y(0)) - (p_g(0) - p_g(0)^k) (p_y(0) - p_y(0)^t)}. \quad (3.25)$$

Una vez resuelto el modelo, la solución proporcionada (es decir, el plan de precontrol) será la óptima para el caso en que δ tome el valor 0. Como se ha mencionado arriba, posteriormente, en la subsección 3.5.3, estudiaremos el comportamiento de $E[n(\delta)]$ como función de δ para dicho plan de precontrol, con el fin de observar si, usando la combinación óptima de (k, t, λ) encontrada para $\delta = 0$, el tamaño muestral podría aumentar mucho en el caso de que el proceso presentase alguna desviación.

3.3.4. Obtención de las restricciones

El modelo (3.1) presenta dos restricciones. La primera de ellas limita el valor que puede tomar la probabilidad de falsa alarma, mientras que la segunda limita la probabilidad de no detectar un proceso descentrado. A continuación vamos a obtener una expresión analítica para ambas restricciones.

La primera restricción,

$$\alpha \leq \alpha^*, \quad (3.26)$$

puede formularse, por (2.20), de la siguiente manera:

$$1 - p_g(0)^k \frac{\sum_{j=0}^{t-1} p_y(0)^j}{1 - \sum_{j=1}^{t-1} p_y(0)^j \sum_{i=1}^{k-1} p_g(0)^i} \leq \alpha^*, \quad (3.27)$$

o bien, haciendo uso de (3.20), como

$$1 - \frac{p_g(0)^k (1 - p_g(0)) (1 - p_y(0)^t)}{(1 - p_g(0)) (1 - p_y(0)) - (p_g(0) - p_g(0)^k) (p_y(0) - p_y(0)^t)} \leq \alpha^*. \quad (3.28)$$

La segunda restricción

$$\beta \leq \beta^* \quad (3.29)$$

indica el valor máximo que debe tomar la probabilidad de cualificar como correcto un proceso que no lo es. Para poder desarrollar esta restricción, es necesario concretar cuándo consideramos que un proceso es o no correcto, ya que dicha probabilidad depende del valor concreto de la desviación, δ . Por ello, tomaremos δ^* una desviación inaceptable del proceso, de manera que, cuando el proceso presente una desviación tal, la probabilidad de que el precontrol termine cualificando el proceso no sea superior a β^* . Teniendo en cuenta esto, la restricción (3.29) se reescribe así:

$$\beta_{\delta^*} \leq \beta^* \quad (3.30)$$

(véase (2.27)).

Es decir, estamos acotando inferiormente en $(1 - \beta^*)$ la potencia del precontrol, visto como contraste sobre la desviación, para una desviación dada δ^* (véase la subsección 2.5.1).

Teniendo en cuenta (2.12) y (3.20), podemos expresar la restric-

ción (3.30) de la siguiente manera:

$$p_g(\delta^*)^k \frac{\sum_{j=0}^{t-1} p_y(\delta^*)^j}{1 - \sum_{j=1}^{t-1} p_y(\delta^*)^j \sum_{i=1}^{k-1} p_g(\delta^*)^i} \leq \beta^* \Leftrightarrow \quad (3.31)$$

$$\frac{p_g(\delta^*)^k (1 - p_g(\delta^*)) (1 - p_y(\delta^*)^t)}{(1 - p_g(\delta^*)) (1 - p_y(\delta^*)) - (p_g(\delta^*) - p_g(\delta^*)^k) (p_y(\delta^*) - p_y(\delta^*)^t)} \leq \beta^*. \quad (3.32)$$

De esta manera, hemos desarrollado analíticamente las restricciones del modelo (3.1).

3.4. El precontrol óptimo

Según lo que acabamos de ver, y para los criterios establecidos anteriormente, el modelo de optimización de planes de precontrol (3.1) queda como sigue:

$$\left. \begin{array}{l} \text{Min} \quad \frac{\sum_{i=0}^{k-1} p_g(0)^i \sum_{j=0}^{t-1} p_y(0)^j}{1 - \sum_{i=1}^{k-1} p_g(0)^i \sum_{j=1}^{t-1} p_y(0)^j} \\ \text{s.a:} \\ 1 - p_g(0)^k \frac{\sum_{j=0}^{t-1} p_y(0)^j}{1 - \sum_{j=1}^{t-1} p_y(0)^j \sum_{i=1}^{k-1} p_g(0)^i} \leq \alpha^* \\ p_g(\delta^*)^k \frac{\sum_{j=0}^{t-1} p_y(\delta^*)^j}{1 - \sum_{j=1}^{t-1} p_y(\delta^*)^j \sum_{i=1}^{k-1} p_g(\delta^*)^i} \leq \beta^* \\ k \geq 1, t \geq 1, \lambda \geq 2 \\ k, t \in \mathbb{Z}, \lambda \in \mathbb{R} \end{array} \right\}, \quad (3.33)$$

o bien, equivalentemente, de esta otra:

$$\left. \begin{array}{l}
 \text{Min} \quad \frac{(1 - p_g(0)^k) (1 - p_y(0)^t)}{(1 - p_g(0)) (1 - p_y(0)) - (p_g(0) - p_g(0)^k) (p_y(0) - p_y(0)^t)} \\
 \text{s.a:} \quad 1 - \frac{p_g(0)^k (1 - p_g(0)) (1 - p_y(0)^t)}{(1 - p_g(0)) (1 - p_y(0)) - (p_g(0) - p_g(0)^k) (p_y(0) - p_y(0)^t)} \leq \alpha^* \\
 \frac{p_g(\delta^*)^k (1 - p_g(\delta^*)) (1 - p_y(\delta^*)^t)}{(1 - p_g(\delta^*)) (1 - p_y(\delta^*)) - (p_g(\delta^*) - p_g(\delta^*)^k) (p_y(\delta^*) - p_y(\delta^*)^t)} \leq \beta^* \\
 k \geq 1, t \geq 1, \lambda \geq 2 \\
 k, t \in \mathbb{Z}, \lambda \in \mathbb{R}
 \end{array} \right\} \cdot \quad (3.34)$$

En este problema, se supone conocida y constante la capacidad potencial del proceso $C_p > 0$ (1.1), o, equivalentemente, σ , la desviación típica del mismo (o una estimación de éstas). Los *parámetros del modelo* son la máxima tasa de falsa alarma aceptada $\alpha^* \in]0, 1[$ (es decir, una cota superior para la probabilidad de no cualificar un proceso con desviación $\delta = 0$), una determinada desviación del proceso $\delta^* > 0$ considerada inaceptable y una cota superior $\beta^* \in]0, 1[$ para la probabilidad de que el procedimiento de precontrol termine sin detectar dicha desviación inaceptable. Nos referiremos al modelo (3.34) como *problema OPC* ($\alpha^*, \beta^*, \delta^*$), o simplemente problema OPC (por *optimal pre-control*).

Como ya hemos destacado, este problema supone un paso más en la dirección marcada por el precontrol generalizado de [San Matías et al. \(2004\)](#): especificado el valor de C_p y fijados por el usuario final unos requerimientos relativos a la potencia del procedimiento (que se concretan en la elección de α^*, β^* y δ^*), éste puede obtener mediante la resolución del problema (3.34) la mejor combinación de los parámetros del precontrol k, t y λ (es decir, el mejor plan de precontrol), de acuerdo con el criterio de minimización del tamaño muestral.

En el problema OPC (3.34), las probabilidades p_g y p_y son funciones

conocidas de la variable del modelo λ (véanse (2.21) y (2.22)), si asumimos la normalidad del proceso y el resto de hipótesis enunciadas al principio de la sección 3.3.

En definitiva, estamos abordando un problema de optimización unicriterio determinista, con dos restricciones, no lineal (tanto en la función objetivo como en las restricciones) y tres variables, una de ellas de naturaleza continua (λ) y las otras dos enteras (k y t). Por lo tanto, el problema (3.34) es un problema de *optimización no lineal entera mixta* (MINLP; véase la subsección 1.4.1). Además, como se demostrará en el capítulo 4 (concretamente, en el corolario 4.6), se trata de un problema no convexo.

En la subsección 1.4.2 se hizo un breve repaso de las técnicas existentes para la resolución de este tipo de problemas. A continuación, evaluaremos mediante una experiencia numérica la posibilidad de aplicación de los diferentes enfoques a la hora de llevar a la práctica y sistematizar la obtención de soluciones al problema OPC (3.34).

3.5. Resolución del modelo del precontrol óptimo. Discusión y experiencia numérica

Hemos caracterizado como problema de MINLP la búsqueda de la mejor configuración de los parámetros del precontrol. El modelo (3.34), una vez resuelto, permitirá al usuario final obtener la configuración de los parámetros k , t y λ que minimice el valor de la función objetivo (3.25), respetando los límites que él imponga para α y β_{δ^*} . Por tanto, la verdadera utilidad del modelo (3.34) consiste en que se pueda resolver de una manera eficaz y eficiente.

Por ello, en esta sección estudiamos las diferentes alternativas que creemos que existen para resolver de modo sistemático y efectivo el problema del precontrol óptimo (3.34), prestando atención no sólo a las técnicas, métodos u algoritmos, sino también a su implementación concreta.

3.5.1. Estrategias y herramientas de resolución

Ante un problema de optimización determinado que se desee resolver, puede elegirse entre aplicar métodos de resolución *genéricos* ya existentes para la clase o las clases dentro de las cuales encaje dicho problema, o bien algoritmos diseñados (o a diseñar) *ad hoc* para éste. Y, en general, tanto en un caso como en el otro, dichos algoritmos pueden estar basados en un enfoque *clásico*, *exacto*, *analítico*, etc. o bien en un enfoque *heurístico* o *metaheurístico*, de *soft computing*.

En la sección 1.4.2 se han enumerado los principales enfoques tanto *clásicos* como *heurísticos* existentes en la literatura para la resolución de problemas de MINLP.

Dichos enfoques dan lugar, por un lado, a algoritmos de propósito más o menos general, en el sentido de que pueden ser aplicados a cualquier problema de MINLP o, según los casos, a algunos subtipos de esta clase de problemas (por ejemplo, [Zhu y Kuno, 2006](#), desarrollan un algoritmo tipo *branch-and-cut* para problemas no lineales convexos *binarios*). Por otro lado, los enfoques mencionados, u otros, pueden ser utilizados también para elaborar algoritmos más específicos orientados a resolver un problema en concreto (es el caso, por citar algún ejemplo, del algoritmo de [Wang y Achenie, 2002](#), que combinan OA con *simulated annealing* para optimizar el diseño de disolventes, o el del trabajo de [Karuppiyah y Grossmann, 2006](#), que adaptan el esquema de *branch-and-bound* espacial al diseño óptimo de sistemas de uso y tratamiento de aguas).

En cualquier caso, hablamos de métodos o algoritmos que finalmente son implementados como *software*, bien orientado a un uso académico o bien con un carácter más comercial, desarrollado y mantenido de manera profesional. Así, un determinado enfoque para la resolución de problemas MINLP (por ejemplo, el B&B) puede dar lugar a diferentes métodos y a diferentes implementaciones de estos métodos (por ejemplo, los algoritmos GAMS/Bonmin B-BB y GAMS/SBB; véase la subsección 3.5.2).

Sendas revisiones recientes del *software* disponible para MINLP pueden encontrarse en los trabajos de [D'Ambrosio y Lodi \(2011\)](#) y [Bussieck y Vigerske \(2011\)](#).

A continuación abordamos la resolución de una batería de ejemplos del problema de precontrol óptimo (3.34) utilizando *software* comercial de propósito general que implementa versiones recientes de los diferentes enfoques para MINLP.

Posteriormente analizaremos los resultados obtenidos, y plantaremos la posibilidad de desarrollar algoritmos específicos para el problema del precontrol óptimo.

3.5.2. Experiencia numérica

En el capítulo anterior hemos mostrado, a través de un ejemplo práctico (véase la sección 2.6), cómo una correcta elección de los parámetros del precontrol permite mejorar el comportamiento de la técnica clásica. En esta subsección retomamos dicho ejemplo, con el fin de poner de manifiesto la utilidad de la propuesta que hemos desarrollado aquí.

Desarrollo de una batería de ejemplos

En el ejemplo al que nos referimos, se buscaba una configuración de los parámetros k , t y λ de manera que el plan de precontrol resultante cumpliera unos ciertos requerimientos (véanse las expresiones (2.35) y (2.36)). Visto a través del nuevo enfoque introducido en el presente capítulo, lo que se estaba haciendo era, sencillamente, *buscar soluciones factibles* del problema de optimización (3.34), con $\alpha^* = 1\%$, $\delta^* = 2,446$ y $\beta^* = 0,5\%$, suponiendo una capacidad potencial C_p de 1,5; de hecho, ya se ha mostrado que existe más de una posible solución (véanse las tablas 2.1 y 2.2).

Nuestra propuesta como problema de optimización permite elegir, de entre todas las posibles configuraciones, la mejor según un determinado criterio. Tomando como punto de partida el ejemplo de la sección 2.6, hemos desarrollado toda una batería de ejemplos (o *instancias*, en una incorrecta adaptación del inglés), para diferentes valores de C_p , α^* , β^* , siempre tomando como desviación inaceptable δ^* aquella que produzca una proporción teórica $p^* = 2\%$ de piezas fuera de especificaciones

(véase la expresión (2.34)), lo cual es bastante razonable. Concretamente, las combinaciones consideradas han sido las siguientes:

- $C_p \in \{1,2; 1,3333; 1,4; 1,5\}$
- $\alpha^* \in \{0,27\%; 0,5\%; 1\%\}$
- δ^* tal que $p^* = 2\%$
- $\beta^* \in \{0,5\%; 5\%; 10\%\}$

Es decir, se han considerado $4 \times 3 \times 3 = 36$ ejemplos distintos que se enumeran en la tabla A.1 del anexo A. Resolveremos el problema del precontrol óptimo (3.34) para cada uno de estos escenarios, que suponen una muestra bastante amplia de los casos de uso más habituales. De acuerdo con lo expuesto en la sección anterior, para la resolución utilizaremos algoritmos de MINLP de propósito general.

Selección de algoritmos a evaluar

Hemos realizado una revisión del *software* comercial de optimización existente que implementa algoritmos de MINLP de uso genérico, centrándonos en las últimas versiones y desarrollos disponibles. Sin descartar ninguna opción que en principio pueda resultar interesante, hemos priorizado en nuestra búsqueda aquellos programas a los que tuviéramos fácil acceso, bien por ser distribuidos gratuitamente, o bien por disponer de alguna licencia de los mismos. El objetivo final no es tanto ser exhaustivo como valorar si es posible resolver de manera eficaz y eficiente el problema del precontrol óptimo usando herramientas ya existentes y que estén al alcance de cualquier potencial usuario.

Tras un primer examen, se decidió considerar los siguientes entornos o programas: LINGO de LINDO Systems ([LINDO Systems Inc., 2013b](#)), GAMS de GAMS Development Corporation ([Rosenthal, 2013](#)) y la herramienta Solver desarrollada por Frontline Systems y que se ejecuta en Microsoft Excel ([Frontline Systems Inc., 2013b](#)). Todos ellos implementan

algoritmos diseñados para resolver modelos de MINLP de cualquier clase, imponiendo pocas restricciones.

En concreto, el entorno GAMS dispone de una amplia colección de *solvers* o módulos³, algunos de ellos de reciente incorporación, que implementan métodos de MINLP de toda índole: basados en B&B, en OA, heurísticos, etc. De hecho, GAMS proporciona acceso a través de uno de sus módulos (concretamente, GAMS/LINDOGlobal) al algoritmo de optimización global no lineal que implementa LINGO (LINDO Global Optimization Procedure, GOP; [LINDO Systems Inc., 2013a](#)). Por ello, finalmente hemos descartado resolver nuestro problema con LINGO, dado que tenemos acceso a su propio algoritmo a través del entorno GAMS.

La tabla 3.1 recoge la lista final de todos los algoritmos con los que hemos abordado la resolución de los 36 ejemplos de nuestro problema (3.34). En ella especificamos, para cada algoritmo, el enfoque concreto en que se basa, si tiene alcance local o global, así como una breve descripción, incluyendo las principales referencias bibliográficas. Además de la información que proporcionamos en esta tabla, para acceder a una revisión más detallada de los algoritmos de MINLP disponibles en GAMS, puede consultarse el trabajo de [Lastusilta \(2011\)](#).

Hay que señalar que se descartaron los *solvers* GAMS/ANTIGONE ([Floudas y Misener, 2013a,b](#)), GAMS/BARON ([Sahinidis, 2013, 1996](#)), GAMS/Couenne ([Vigerske, 2013; Belotti et al., 2009](#)) y GAMS/SCIP ([GAMS Development Corp., 2013d](#)) debido a que, a pesar de estar diseñados para la resolución de problemas de MINLP, en sus versiones actuales no admiten modelos que involucren el uso de la función de distribución acumulada de la distribución normal tipificada Φ^4 , la cual forma parte importante de nuestro modelo, como ya es sabido. Los cuatro algoritmos descartados se fundamentan en una estrategia de búsqueda

³GAMS está abierto a la colaboración de terceros que incorporan al catálogo de algoritmos disponibles sus propias aportaciones.

⁴En el sistema de modelización del entorno GAMS, la función de distribución Φ es implementada de manera interna mediante la función `errorf` ([Rosenthal, 2013](#)).

global de tipo B&B. Concretamente, GAMS/BARON, GAMS/Couenne y GAMS/SCIP implementan un *branch-and-bound* espacial; por su parte, GAMS/ANTIGONE es una extensión del algoritmo para problemas cuadráticos GloMIQO (Global Mixed-Integer Quadratic Optimizer; Misener y Floudas, 2013) que utiliza relajaciones convexas y técnicas de preproceso.

Algunos de los algoritmos de MINLP disponibles en GAMS y descritos en la tabla 3.1, debido a su enfoque y a su implementación, precisan a su vez hacer uso auxiliar de algoritmos de MILP y/o de NLP. En tales casos, hemos probado dichos algoritmos de MINLP con cada una de las configuraciones resultantes de considerar como algoritmos auxiliares los diferentes *solvers* para problemas de MILP y NLP disponibles en GAMS y admitidos por cada *solver*. Concretamente, en las tablas 3.2 y 3.3 se detallan los *solvers* de MILP y de NLP, respectivamente, admitidos por cada *solver* de MINLP de GAMS en su versión actual. El interés está justificado, ya que, según destacan D'Ambrosio y Lodi (2011), la elección de un buen método auxiliar de MILP y/o NLP influye en el rendimiento final del método de MINLP. Por lo demás, en todos los casos se han respetado las configuraciones básicas por defecto de cada algoritmo (tanto de los principales como de los auxiliares); profundizamos en este aspecto en el apartado siguiente. En consecuencia, y teniendo en cuenta todo lo anterior, el número total de variantes distintas de algoritmos evaluadas en esta experiencia numérica ha sido 135 (véase el detalle en la tabla 3.4).

Configuración y condiciones iniciales de los algoritmos

Como es sabido, y hablando en términos generales, los algoritmos de resolución de problemas no lineales (enteros o no) pueden converger a soluciones finales distintas, o incluso no converger, en función de cuál sea el punto inicial (o el conjunto de puntos iniciales, en el caso de algoritmos basados en poblaciones) a partir del cual comienzan a iterar. En esta experiencia numérica se decidió tomar como punto de partida para todos los algoritmos probados la solución correspondiente al diseño clásico del

Algoritmo	Enfoque [*]	Alcance [†]	Descripción e información adicional
Excel Solver	B&B	Global	<p>Solver implementa un algoritmo tipo <i>branch-and-bound</i> para problemas de MINLP. Para resolver los subproblemas de NLP utiliza una versión del algoritmo del gradiente reducido generalizado (Lasdon et al., 1978) implementada y adaptada por sus autores de manera conjunta con Frontline Systems.</p> <p>Versión: La incluida en Excel 2010.</p> <p>Autores: L.S. Lasdon y A.D. Waren, University of Texas at Austin y Cleveland State University; y Frontline Systems, Inc.</p> <p>Principales referencias: Frontline Systems Inc. (2013b), Frontline Systems Inc. (2013a), Fylstra et al. (1998), Lasdon et al. (1978).</p>
GAMS/AlphaECP	ECP	Local (ext.)	<p>Garantiza el óptimo global para problemas de MINLP pseudoconvexos. Aplica un <i>solver</i> auxiliar de NLP para refinar las soluciones obtenidas por el <i>solver</i> auxiliar de MILP. En problemas no convexos permite usar un heurístico que reselecciona planos de corte. Diseñado por los autores originales de la técnica ECP.</p> <p>Versión: 2.10.02.</p> <p>Autores: T. Westerlund y K. Lundqvist, Åbo Akademi University.</p> <p>Principales referencias: Westerlund y Lastusilta (2013), Westerlund y Pörn (2002).</p>

Tabla 3.1: Relación de los algoritmos considerados en la experiencia numérica para resolver la batería de ejemplos del problema del precontrol óptimo. *Continúa en la página siguiente.*

Algoritmo	Enfoque [*]	Alcance [†]	Descripción e información adicional
GAMS/Bonmin B-BB	B&B	Local	<p><i>Branch-and-bound</i> sencillo. El algoritmo es exacto para problemas de MINLP convexos. Al igual que el resto de <i>solvers</i> de Bonmin, forma parte del proyecto de código abierto COIN-OR (The COIN-OR Foundation Inc., 2012).</p> <p>Versión: 1.7.2.</p> <p>Autores: P. Bonami et al., Carnegie Mellon University e IBM Research.</p> <p>Principales referencias: Vigerske (2013), Bonami y Lee (2013), Bonami et al. (2008), Gupta y Ravindran (1985).</p>
GAMS/Bonmin B-ECP	QG	Local	<p>Variante del algoritmo de QG que incorpora cortes del estilo del método del ECP. Toma ideas del algoritmo FilMINT (Abhishek et al., 2010a).</p> <p>Versión: 1.7.2.</p> <p>Autores: P. Bonami et al., Carnegie Mellon University e IBM Research.</p> <p>Principales referencias: Vigerske (2013), Bonami y Lee (2013), Bonami et al. (2008).</p>
GAMS/Bonmin B-OA	OA	Local	<p>Implementación sencilla de la técnica de OA de Duran y Grossmann (1986).</p> <p>Versión: 1.7.2.</p> <p>Autores: P. Bonami et al., Carnegie Mellon University e IBM Research.</p> <p>Principales referencias: Vigerske (2013), Bonami y Lee (2013), Bonami et al. (2008), Duran y Grossmann (1986), Fletcher y Leyffer (1994).</p>

Tabla 3.1: Relación de los algoritmos considerados en la experiencia numérica para resolver la batería de ejemplos del problema del precontrol óptimo. *Continúa en la página siguiente.*

Algoritmo	Enfoque [*]	Alcance [†]	Descripción e información adicional
GAMS/Bonmin B-QG	QG	Local	Implementación sencilla del algoritmo QG de Quesada y Grossmann (1992) . Versión: 1.7.2. Autores: P. Bonami et al., Carnegie Mellon University e IBM Research. Principales referencias: Vigerske (2013) , Bonami y Lee (2013) , Bonami et al. (2008) , Quesada y Grossmann (1992) .
GAMS/Bonmin B-Hyb	B&B + QG	Local	Algoritmo híbrido entre el B-BB y el B-QG arriba descritos. En cada nodo de la estrategia <i>branch-and-bound</i> , el algoritmo decide si resolver un problema no lineal (B-BB) o un problema lineal (B-QG). Versión: 1.7.2. Autores: P. Bonami et al., Carnegie Mellon University e IBM Research. Principales referencias: Vigerske (2013) , Bonami y Lee (2013) , Bonami et al. (2008) .
GAMS/Bonmin B-iFP	FP + OA	Local	Implementa la <i>iterated feasibility pump</i> (bomba de factibilidad iterada), un algoritmo especializado en la búsqueda de soluciones factibles de problemas convexos de MINLP mediante la resolución alternante de problemas de NLP y MILP, y que posteriormente es iterado para mejorar la solución factible inicial, integrándose en un esquema de aproximación exterior. Diseñado por los autores originales de este algoritmo. Autores: P. Bonami et al., Carnegie Mellon University e IBM Research. Principales referencias: Vigerske (2013) , Bonami y Lee (2013) , Bonami et al. (2008) , Bonami et al. (2009) .

Tabla 3.1: Relación de los algoritmos considerados en la experiencia numérica para resolver la batería de ejemplos del problema del precontrol óptimo. *Continúa en la página siguiente.*

Algoritmo	Enfoque [*]	Alcance [†]	Descripción e información adicional
GAMS/DICOPT	OA	Local (ext.)	<p>Implementa la técnica de aproximación exterior, extendida para tratar restricciones de igualdad y restricciones no convexas, si bien no garantiza alcanzar el óptimo global en el caso no convexo.</p> <p>Versión: 24.1.3.</p> <p>Autores: J. Viswanathan e I.E. Grossmann, Carnegie Mellon University.</p> <p>Principales referencias: Grossmann et al. (2013), Duran y Grossmann (1986), Kocis y Grossmann (1987), Kocis y Grossmann (1989), Viswanathan y Grossmann (1990).</p>
GAMS/KNITRO	B&B	Local	<p>Algoritmo que implementa tres métodos propios de NLP para la búsqueda de óptimos locales, de los cuales elige automáticamente el más adecuado para cada ocasión. Es considerado robusto y eficaz, también en problemas de gran tamaño. Se trata de la versión para GAMS de un <i>software</i> comercial desarrollado por Ziena Optimization LLC (Ziena Optimization LLC, 2013b).</p> <p>Versión: 8.1.1.</p> <p>Autores: R.A. Waltz et al, Ziena Optimization LLC.</p> <p>Principales referencias: Ziena Optimization LLC (2013a), Byrd et al. (2006).</p>

Tabla 3.1: Relación de los algoritmos considerados en la experiencia numérica para resolver la batería de ejemplos del problema del precontrol óptimo. *Continúa en la página siguiente.*

Algoritmo	Enfoque [*]	Alcance [†]	Descripción e información adicional
GAMS/LINDOGlobal	B&B	Global	<p>Versión para GAMS del algoritmo Global Optimization Procedure (GOP) de LINGO (LINDO Systems Inc., 2013b). Para encontrar el óptimo global, se comprueba la convexidad de cada subproblema generado por la estrategia de <i>branch-and-bound</i>, subdividiéndolo a su vez en el caso de que no sea convexo.</p> <p>Versión: 24.1.3. Autores: LINDO Systems Inc. Principales referencias: LINDO Systems Inc. (2013a), Lin y Schrage (2009), LINDO Systems Inc. (2013b).</p>
GAMS/OQNLP	OQNLP	Global (heur.)	<p>Algoritmo heurístico con método multiarranque. Los diferentes puntos iniciales son determinados utilizando búsqueda dispersa (Glover, 1977; Laguna y Marti, 2003), y posteriormente son mejorados mediante un algoritmo auxiliar de NLP.</p> <p>Versión: 24.1.3. Autores: Optimal Methods Inc. Principales referencias: Optimal Methods Inc. y OptTek Systems Inc. (2013), Ugray et al. (2005).</p>

Tabla 3.1: Relación de los algoritmos considerados en la experiencia numérica para resolver la batería de ejemplos del problema del precontrol óptimo. *Continúa en la página siguiente.*

Algoritmo	Enfoque [*]	Alcance [†]	Descripción e información adicional
GAMS/SBB	B&B	Local	Implementación sencilla del <i>branch-and-bound</i> que permite utilizar alguno de los solvers NLP de GAMS para resolver los subproblemas no lineales en cada nodo. Implementa una estrategia basada en pseudo-costes para decidir por qué variable bifurcar. Es, tras GAMS/DICOPT, el segundo <i>solver</i> que incorporó GAMS para problemas de MINLP. Versión: 24.1.3. Autores: GAMS Development Corporation. Principales referencias: GAMS Development Corp. (2013c) , Gupta y Ravindran (1985) .

^{*} B&B = Bifurcación y acotación. ECP = Plano de corte extendido. FP = *Feasibility pump*. OA = Aproximación exterior. OQNLP = OptQuest/NLP. QG = *Branch-and-bound* no lineal de Quesada y Grossmann. Todos estos enfoques han sido presentados en la sección 1.4.2.

[†] Local = Algoritmo exacto diseñado para problemas convexos. Local (ext.) = Algoritmo exacto para problemas convexos que contempla heurísticas para el caso no convexo. Global = Algoritmo de optimización global. Global (heur.) = Algoritmo heurístico con vocación global.

Tabla 3.1: Relación de los algoritmos considerados en la experiencia numérica para resolver la batería de ejemplos del problema del precontrol óptimo.

Algoritmo auxiliar MILP	Algoritmos MINLP de GAMS a probar*		
	AlphaECP, DICOPT	Bommin B-OA, Bommin B-iFP	El resto
GAMS/BDMLP (GAMS Development Corp., 2013a)	✓		
GAMS/CBC (Vigerske, 2013)	✓	✓	
GAMS/CPLEX (GAMS Development Corp., 2013b)	✓		
GAMS/GUROBI (Gurobi Optimization Inc., 2013)	✓		
GAMS/MOSEK (MOSEK ApS, 2013)	✓		
GAMS/SULUM (Sulum Optimization, 2013)	✓		
GAMS/XA (Sunset Software Technology, 2013)	✓		
GAMS/XPRESS (FICO, 2013)	✓		
No requiere <i>solver</i> de MILP o usa uno interno.			✓
— * Véase la relación completa de algoritmos en la tabla 3.1.			

Tabla 3.2: Algoritmos de programación lineal entera admitidos por los diferentes *solvers* para MINLP de GAMS.

precontrol, es decir:

$$\text{Solución inicial} = (k = 5, t = 2, \lambda = 4). \tag{3.35}$$

Además, se optó por mantener todas las configuraciones por defecto de los algoritmos, tanto en el caso de Excel como en el de GAMS. Al respecto de las opciones de configuración comunes a todos los algoritmos en GAMS, destacamos que se respetó la configuración predeterminada que fija el número máximo de iteraciones en $2 \cdot 10^9$ y el uso máximo de tiempo de la CPU (tiempo efectivo de proceso) en 1000 segundos⁵.

⁵Opciones *iterlim* y *reslim* de GAMS, respectivamente (McCarl et al., 2013).

Algoritmo auxiliar NLP	Algoritmos MINLP de GAMS a probar*		
	AlphaECP, DICOPT, SBB, OQNLP	Todos los Bommin	El resto
GAMS/CONOPT (Drud, 2013)	✓		
GAMS/IPOPT (Vigerske, 2013)	✓	✓	
GAMS/LGO (Pintér, 2013)	✓		
GAMS/MINOS (Murtagh et al., 2013)	✓		
GAMS/MSNLP (LSGRG) (Optimal Methods Inc. y OptTek Systems Inc., 2013)	✓		
GAMS/PATHNLP (Ferris, 2013)	✓		
GAMS/SNOPT (Gill et al., 2013)	✓		
No requiere <i>solver</i> de NLP o usa uno interno.			✓

—
* Véase la relación completa de algoritmos en la tabla 3.1.

Tabla 3.3: Algoritmos de programación no lineal admitidos por los diferentes *solvers* para MINLP de GAMS.

En cualquier caso, el criterio ha sido siempre el mismo: respetar todas las opciones establecidas por defecto. Esto es así porque entre los objetivos de esta experiencia numérica está comprobar qué resultados obtendría un usuario no avanzado al intentar resolver el problema OPC (3.34) mediante *software* de optimización de propósito general.

Adaptación del modelo a los entornos de resolución seleccionados

La adaptación del modelo (3.34) a la sintaxis de Excel y GAMS ha resultado no excesivamente complicada pero tampoco trivial: en ambos entornos se comprobó mediante pruebas preliminares que la manera concreta de codificar las diferentes expresiones que intervienen

Algoritmos MINLP a probar	Total combi- naciones		<i>Solvers</i> MILP aux.		<i>Solvers</i> NLP aux.
Excel Solver	1				
GAMS/AlphaECP	56	=	8	×	7
GAMS/Bonmin B-BB	1	=			1
GAMS/Bonmin B-ECP	1	=			1
GAMS/Bonmin B-OA	1	=	1	×	1
GAMS/Bonmin B-QG	1	=			1
GAMS/Bonmin B-Hyb	1	=			1
GAMS/Bonmin B-iFP	1	=	1	×	1
GAMS/DICOPT	56	=	8	×	7
GAMS/KNITRO	1				
GAMS/LINDOGlobal	1				
GAMS/OQNLP	7	=			7
GAMS/SBB	7	=			7
Total	135				

Tabla 3.4: Cálculo del número total de variantes de algoritmos de MINLP a evaluar en la experiencia numérica de la subsección 3.5.2.

en el modelo condicionaba no sólo la velocidad de convergencia de los algoritmos sino también, en ocasiones, el hecho de si convergían o no⁶. La conclusión que puede extraerse es que, en el uso de *software* comercial para la resolución de un modelo de optimización, la adaptación del modelo a la sintaxis y las condiciones de funcionamiento de dicho *software* requiere un tiempo y un esfuerzo de análisis no despreciables; incluso ejemplos diferentes del mismo modelo (en nuestro caso, cualquiera de los 36 ejemplos que hemos considerado u otros que se puedan

⁶Entre otras cosas, se probó la utilización las expresiones originales del modelo (3.33) o las alternativas del (3.34); asimismo, se exploró el uso de expresiones auxiliares para representar cálculos que se repiten en diferentes partes del modelo. Por otro lado, en GAMS la restricción original $\lambda \geq 2$ tuvo que ser sustituida por $\lambda \geq 2,001$ para evitar problemas de estabilidad computacional de algunos algoritmos en puntos demasiado cercanos a $\lambda = 2$.

tener en cuenta) pueden precisar de adaptaciones concretas diferentes para asegurar el correcto funcionamiento del programa sobre ellos. Principalmente esto último nos hace pensar que, en ocasiones, el uso de *software* comercial de propósito general para la resolución repetida bajo diferentes condiciones iniciales de problemas optimización por parte de usuarios no avanzados puede no ser la opción más recomendable.

Una adaptación importante requerida por prácticamente todos los algoritmos considerados para poder tratar con nuestro modelo, y que de hecho es bastante lógica, es la incorporación de respectivas *cotas superiores* para las variables k , t y λ , además de las cotas inferiores ya establecidas (véase la subsección 3.2.1). Incluso algunos de ellos fijan dichas cotas por su cuenta en el caso de detectar variables que no estén ya acotadas dentro de un cierto intervalo⁷. Tras realizar algunas pruebas iniciales, se fijó un valor de 41 como cota superior tanto para k como para t , y una cota superior para λ de 42 (de manera que la diferencia entre el máximo y el mínimo valor admitidos para cada variable fuese en todos los casos igual a 40).

Por el significado que tienen dichas variables en el contexto del problema que estamos tratando de resolver, las cotas superiores elegidas son suficientemente grandes como para no suponer una limitación real a la hora de encontrar la solución óptima. Por un lado, asumiendo que el precontrol se trata de un método para valorar de manera rápida un proceso, no son razonables en la práctica configuraciones que obliguen a observar más de 41 piezas verdes o amarillas consecutivas para tomar una decisión. Además, dado que el objetivo de nuestro problema es minimizar el tamaño muestral esperado, es razonable pensar que el óptimo se obtendrá para valores relativamente pequeños de k y t . Por otro lado, soluciones con valores de λ superiores a 42 supondrían que la anchura de la zona verde (2.24) debería reducirse a menos de la veintiunava parte de la anchura del rango de tolerancias, lo cual tiene poca aplicación real.

⁷De hecho, la existencia de problemas de MINLP *indecidibles* (es decir, que no admiten una solución algorítmica) está asociada a la presencia de regiones factibles no acotadas (Burer y Letchford, 2012).

En consecuencia, hacemos notar que los ejemplos que estamos resolviendo numéricamente aquí *no* corresponden exactamente al problema OPC (3.34) sino a su versión *acotada*, que llamaremos *problema OPC-b* (α^* , β^* , δ^* , $K_{\text{máx}}$, $T_{\text{máx}}$, $L_{\text{máx}}$), o simplemente problema OPC-b (por *bounded*):

$$\left. \begin{array}{l}
 \text{Min} \quad \frac{(1 - p_g(0)^k) (1 - p_y(0)^t)}{(1 - p_g(0)) (1 - p_y(0)) - (p_g(0) - p_g(0)^k) (p_y(0) - p_y(0)^t)} \\
 \text{s.a:} \quad 1 - \frac{p_g(0)^k (1 - p_g(0)) (1 - p_y(0)^t)}{(1 - p_g(0)) (1 - p_y(0)) - (p_g(0) - p_g(0)^k) (p_y(0) - p_y(0)^t)} \leq \alpha^* \\
 \frac{p_g(\delta^*)^k (1 - p_g(\delta^*)) (1 - p_y(\delta^*)^t)}{(1 - p_g(\delta^*)) (1 - p_y(\delta^*)) - (p_g(\delta^*) - p_g(\delta^*)^k) (p_y(\delta^*) - p_y(\delta^*)^t)} \leq \beta^* \\
 1 \leq k \leq K_{\text{máx}}, 1 \leq t \leq T_{\text{máx}}, 2 \leq \lambda \leq L_{\text{máx}} \\
 k, t \in \mathbb{Z}, \lambda \in \mathbb{R}
 \end{array} \right\}, \quad (3.36)$$

para una capacidad potencial $C_p > 0$ conocida, cumpliendo $K_{\text{máx}}, T_{\text{máx}} \in \mathbb{Z}, K_{\text{máx}}, T_{\text{máx}} \geq 1$ y $L_{\text{máx}} \geq 2$ (en este caso $K_{\text{máx}} = T_{\text{máx}} = 41$ y $L_{\text{máx}} = 42$), y estando los parámetros α^* , β^* y δ^* delimitados como se especificó al definir el problema OPC original (3.34). De hecho, mantendremos este planteamiento *acotado* cuando abordemos en el capítulo 5 el desarrollo de algoritmos específicos para el problema del precontrol óptimo.

Además de adaptar el modelo a las sintaxis de cada sistema, hemos hecho uso de las funcionalidades para programación de rutinas existentes en GAMS y en Excel para ejecutar de manera secuencial y automática todos los algoritmos seleccionados sobre todos los ejemplos considerados.

Resultados numéricos

Los algoritmos se han ejecutado sobre la batería de ejemplos utilizando las versiones 2010 de Excel y 24.1.3 de GAMS (de julio de 2013). Las ejecuciones se han realizado en un ordenador con procesador Intel Core i5-650 de dos núcleos y con una velocidad de 3,20 GHz, provisto de 3,86 GB de memoria RAM utilizable y sistema operativo Windows 7 a 64 bits.

Básicamente, Solver puede terminar su ejecución informando de haber encontrado una solución óptima (supuestamente en sentido global, aunque en algún caso no tiene por qué ser así) que satisface las restricciones, o detectando que el problema es no acotado o que es no factible, o bien puede finalizar de manera inesperada debido a algún tipo de error (habitualmente causado por soluciones intermedias que exceden el dominio de alguna función del problema, divisiones por cero, etc.). En nuestra batería de ejemplos se observaron todos los casos excepto el del problema no acotado⁸. Además, durante las pruebas preliminares ya aludidas, y que se volverán a mencionar más adelante, se observaron disparidades en la solución final obtenida mediante esta herramienta en función de la solución inicial proporcionada, a pesar de ser catalogada como un método de optimización de alcance global.

Por su parte, GAMS proporciona información bastante precisa sobre el *estado* del modelo tras su resolución y del algoritmo aplicado para ello. De entre los estados posibles⁹, los que se han producido al resolver la batería de ejemplos del precontrol óptimo con los diferentes algoritmos de GAMS pueden agruparse en los siguientes casos:

1. Se finaliza el algoritmo bien porque ha terminado su ejecución o bien porque se ha alcanzado el tiempo máximo de ejecución fijado, y se obtiene una solución óptima entera, *posible óptimo (local)*, pero no se asegura que lo sea (en ninguno de los dos casos).
2. Se finaliza el algoritmo bien porque ha terminado su ejecución o bien porque se ha alcanzado el tiempo máximo de ejecución fijado, concluyendo que el problema es *localmente no factible*, o al menos devolviendo una solución no entera (no factible, por tanto); es decir,

⁸En el capítulo 4 se estudiarán las propiedades teóricas de las funciones que intervienen en el problema OPC (3.34) y en su versión acotada (3.36), pero es claro que ninguno de ellos puede ser no acotado, ya que su función objetivo (número esperado de piezas a muestrear) está acotada inferiormente por 1, por definición.

⁹El resultado de ejecutar cualquier *solver* de GAMS sobre un problema dado se codifica en forma de 19 posibles estados para el modelo (*model status*), algunos de ellos asociados sólo con determinadas clases de problemas, y 13 posibles estados para el *solver* (*solver status*). Pueden consultarse en Rosenthal (2013).

en la zona explorada partiendo de la solución inicial proporcionada el algoritmo no ha sido capaz de encontrar soluciones factibles, pero *no se garantiza que no las haya* en otras zonas del espacio de búsqueda no exploradas.

3. Se finaliza de manera inesperada por algún tipo de error.

Hay que destacar, por tanto, que ninguno de los *solvers* de GAMS probados finalizó garantizando haber resuelto ninguno de los problemas de la batería de forma (al menos) localmente óptima. Esto es quizás debido a que se tuvo que excluir de la prueba a casi todos los algoritmos de MINLP con vocación global disponibles en GAMS, como se ha mencionado anteriormente, y que el único algoritmo no heurístico global probado, GAMS/LINDOGlobal, finalizó su ejecución por consumir el tiempo máximo fijado, en todos los casos. En nuestro contexto, este hecho en sí es relativamente poco importante, pues tiene que ver únicamente con la forma en que GAMS clasifica el resultado de la resolución de un modelo¹⁰.

En cualquier caso, en la práctica todo lo anterior significa que, al menos sobre nuestra batería de ejemplos, *todos los algoritmos probados (incluso los que supuestamente tienen alcance global) se han comportado como heurísticos*, es decir, no garantizan la optimalidad ni la infactibilidad¹¹.

Descripción del comportamiento de los problemas En la tabla A.1 del anexo A se muestra la mejor solución encontrada para cada uno de los

¹⁰De hecho, tal y como se indica en el manual de uso de GAMS (Rosenthal, 2013), el estado *optimal* (solución óptima global hallada) se reserva sólo para el caso lineal. Para problemas no lineales GAMS, en el mejor de los casos, informará de que se ha alcanzado una solución localmente óptima, y es el usuario quien tienen que comprobar la convexidad del problema para, en su caso, determinar que se trata de un óptimo global.

¹¹Como ya se ha mencionado al final de la sección 3.4 y se demostrará en el capítulo 4, el problema OPC (3.34) es no convexo, lo cual hace esperar que todos los algoritmos enfocados a problemas convexos tengan un comportamiento heurístico sobre él (véase la subsección 1.4.2). Además, esta experiencia numérica nos muestra que incluso los algoritmos supuestamente capaces de encontrar el óptimo global de nuestro problema parecen no lograrlo, al menos no en un tiempo razonable.

ejemplos de la batería. Además, se indica el porcentaje de algoritmos que alcanzaron dicha solución, así como el porcentaje de algoritmos que no produjo ninguna (bien por concluir con infactibilidad o bien con error) y el de aquellos que terminaron con error. Esto proporciona una primera idea de qué problemas han sido más fáciles o difíciles de resolver. En la tabla 3.5 mostramos un resumen de dicha información.

	Mín.–Máx.	Q1–Q3 [‡]	Mediana
Ejecuciones que no devuelven ninguna solución* (sólo problemas factibles [†]) (%)	1–57	7–10	7
Ejecuciones que terminan con error (%)	0–7	1–1	1
Ejecuciones que alcanzan la mejor solución obtenida (sólo problemas factibles [†]) (%)	30–85	50–65	58
Ejecuciones que alcanzan la mejor solución obtenida, respecto de las que proporcionan una solución (%)	38–87	56–71	63

—

* Engloba las experiencias que terminan con error del algoritmo y aquellas en que éste determina que el problema es (localmente) no factible.

† Hay 5 problemas presumiblemente no factibles (ningún algoritmo devuelve ninguna solución).

‡ Primer y tercer cuartil, respectivamente.

Tabla 3.5: Resumen del comportamiento de las ejecuciones realizadas sobre la batería de problemas en la experiencia numérica de la subsección 3.5.2, desde el punto de vista de los problemas.

Para los problemas números 1 a 5 (todos con $C_p = 1,2$), ninguno de los algoritmos devolvió ninguna solución, concluyendo todos que el problema en cuestión era (localmente) no factible o bien devolviendo algún tipo de error.

En los 31 problemas restantes hubo siempre algún algoritmo que no fue capaz de devolver ninguna solución, variando desde el ejemplo número 7 ($C_p = 1,2$, $\alpha^* = 0,01$, $\beta^* = 0,005$), para el que un un 57% de los 135 algoritmos no logró hallar ninguna solución, hasta el ejemplo

número 29 ($C_p = 1,5$, $\alpha^* = 0,0027$, $\beta^* = 0,05$) con sólo un 1 % de ellos.

De hecho, los 4 problemas con $C_p = 1,2$ para los que sí se encontró alguna solución están entre los 5 ejemplos de la batería con mayor porcentaje de algoritmos que fallaron en la búsqueda de alguna solución, variando éste desde el 16 al ya mencionado 57 %. Se intuye que esto es debido a que, sin ser no factibles, estos problemas presentan relativamente pocas soluciones posibles, de modo que algunos algoritmos tienen más dificultades para encontrar al menos una de ellas, partiendo de la solución inicial dada.

Excluyendo todos los ejemplos con $C_p = 1,2$ (no sólo los 5 presumiblemente no factibles, sino también los otros 4), la proporción de algoritmos que no alcanzan a devolver una solución, en cada problema, varía desde el 1 hasta el 16 %; la mediana se mantiene en el 7 %. De nuevo, dentro de esta selección, son los problemas más restrictivos aquellos que, generalmente, tienen asociado un mayor porcentaje de algoritmos que no devuelven ninguna solución para ellos. En concreto, hay 4 problemas con al menos un 12 % de algoritmos sin solución, mientras que, en el resto, el porcentaje es igual o inferior al 9 % (con primer y tercer cuartiles iguales a 7 %). Esos 4 problemas son los ejemplos 13, 17, 26 y 34. Todos ellos tienen en común un valor para β^* de 0,005 o 0,05 (se da la circunstancia de que, en la totalidad de ejemplos estudiados con solución, la restricción relativa al error de tipo II es activa —es decir, se cumple con igualdad— cuando se aplica la mejor configuración encontrada para los parámetros del precontrol).

El problema en el que más algoritmos alcanzaron la mejor solución fue de nuevo el ejemplo 29 (un 85 % de los 135 algoritmos probados, o un 86 % sobre el total de algoritmos que proporcionaron alguna solución para dicho problema).

Por otro lado, el que mayor número de errores generó fue el ejemplo 34 ($C_p = 1,5$, $\alpha^* = 0,01$, $\beta^* = 0,005$), concretamente en un 7 % de los algoritmos.

En cuanto a las mejores soluciones halladas, todas presentan unos valores razonables tanto para las variables k , t y λ como para el mínimo de

la función objetivo, excepto quizás la solución del caso 7 ya mencionado, que establece que hay que llegar a observar $k = 11$ piezas verdes consecutivas para cualificar el proceso, o bien $t = 5$ amarillas consecutivas para rechazarlo, con un valor esperado de 27,84 piezas a muestrear para finalizar el precontrol cuando el proceso esté bajo control.

Estos datos parecen indicar que, en general, los problemas más difíciles de resolver (en los que menos algoritmos han tenido éxito y menos han sido capaces de alcanzar la mejor solución) son aquellos con un valor inferior para el parámetro correspondiente a la capacidad del proceso C_p , es decir, aquellos en los que la variabilidad del proceso es más alta en relación a las tolerancias.

De igual forma se confirma lo que intuitivamente es cierto: con condiciones menos restrictivas se obtienen mejores soluciones. Esto se aprecia claramente en la figura 3.1, que muestra el mejor (es decir, menor) valor de la función objetivo encontrado para cada uno de los ejemplos de la batería. En la gráfica se observa cómo, a medida que los valores de C_p , α^* y β^* se hacen más favorables o menos restrictivos (esto es, valores mayores de C_p , equivalentes a menor variabilidad del proceso, y valores mayores para los límites superiores α^* y β^*), es posible encontrar configuraciones del precontrol que verifiquen las restricciones utilizando un menor número esperado de piezas a muestrear, en el caso de que el proceso esté bajo control.

Descripción del comportamiento de los algoritmos De manera paralela a este análisis de los problemas de la batería, en la tabla A.2 del mismo anexo se relacionan los 135 algoritmos probados y se indica, para cada uno de ellos, el porcentaje de veces que ha alcanzado la mejor solución, así como el porcentaje de veces en que no ha proporcionado ninguna solución válida (por infactibilidad o por error) y el porcentaje de casos en que ha terminado con error. También se indica el porcentaje de ejecuciones en que el algoritmo ha finalizado consumiendo el tiempo máximo asignado. Por último, la tabla presenta medidas sobre la calidad de las soluciones proporcionadas por cada algoritmo, que serán introducidas más adelante.

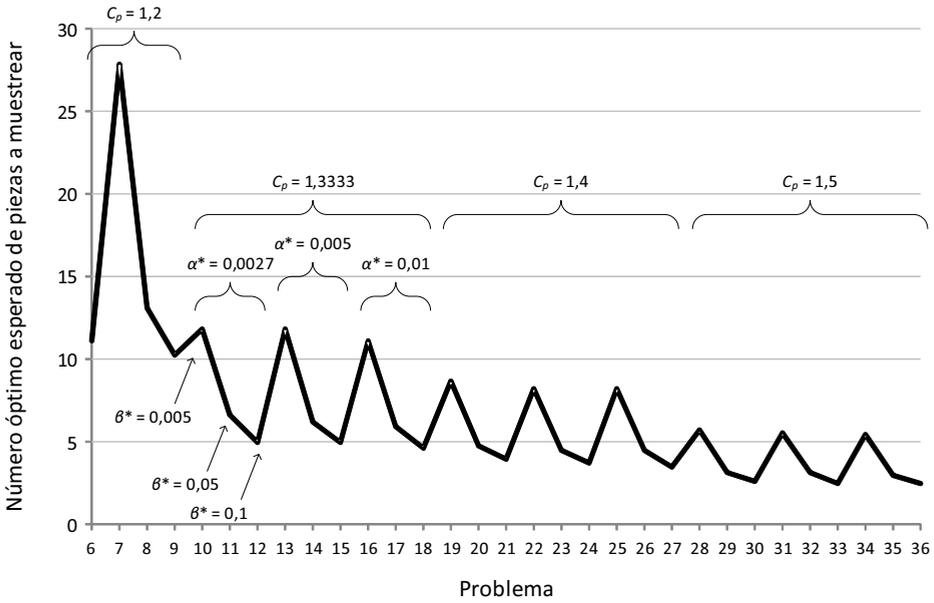


Figura 3.1: Mínimo número esperado de piezas a muestrear para finalizar el procedimiento de precontrol cuando el proceso está centrado, obtenido para cada ejemplo de la batería mediante los algoritmos probados en la experiencia numérica de la subsección 3.5.2. Se excluyen los ejemplos 1 a 5, ya que para ellos ningún algoritmo fue capaz de encontrar ninguna solución.

De este modo, puede realizarse un primer análisis descriptivo de qué algoritmos han tenido un mejor o peor comportamiento sobre la subbatería formada por los ejemplos que se tiene garantía de que son factibles (del problema 6 al 36).

En concreto, 40 (29,6 %) de los 135 algoritmos probados devolvieron alguna solución factible en las 31 ejecuciones que estamos considerando para cada uno de ellos (todas excepto las realizadas sobre los ejemplos supuestamente infactibles, del 1 al 5); y 124 (91,9%) devolvieron solución en al menos el 80 % de las ejecuciones. Las 8 variantes de GAMS/DICOPT que usaron GAMS/MSNLP como *solver* auxiliar de NLP no lograron devolver una solución válida en el 97% de sus respectivas ejecuciones.

Como se ha explicado arriba, la ausencia de una solución válida tras

la ejecución de un algoritmo sobre un problema concreto de la batería pudo deberse simplemente a que el algoritmo devolvió una solución no factible, o bien a que experimentó un error de ejecución. La gran mayoría de los algoritmos (124, un 91,9 %; pero no exactamente los mismos 124 nombrados arriba) terminaron sin error todas de sus ejecuciones. Destacan por su comportamiento *patológico* el algoritmo GAMS/Bonmin B-ECP y la herramienta Solver de Excel, con un porcentaje de finalizaciones con error del 45 % y el 68 %, respectivamente.

Un total de 115 (85,2 %) algoritmos finalizaron sin error todas sus ejecuciones y devolvieron además una solución factible en al menos el 80 % de ellas.

Un total de 6 (4,4 %) algoritmos, todos ellos variantes de GAMS/DICOPT con GAMS/SNOPT como *solver* de NLP, devolvieron la mejor solución conocida para los 31 problemas (véase el detalle en la tabla 3.6); y hasta 48 (35,6 %) lo hicieron en al menos el 84 % de las ejecuciones. De estos 48 algoritmos, 47 son variantes de GAMS/DICOPT y el otro es GAMS/Bonmin B-BB. En el otro costado, un grupo de 12 (8,9 %) algoritmos alcanzaron la mejor solución conocida en menos del 10 % de las ejecuciones; entre ellos, las 8 variantes de GAMS/DICOPT con GAMS/MSNLP como método de NLP, ya mencionadas antes, junto con los algoritmos GAMS/Bonmin B-QG y GAMS/Bonmin B-iFP, los cuales, si bien devolvieron una solución válida en casi todas sus respectivas ejecuciones (en el 97 y el 94 % de ellas, en uno y otro caso), nunca alcanzaron la mejor conocida.

GAMS/Bonmin B-Hyb y de nuevo GAMS/Bonmin B-iFP son los que más veces devolvieron como solución a un problema la peor devuelta por los algoritmos probados; concretamente, en el 35 y el 42 % de sus respectivas ejecuciones.

Por último, destaca el hecho de que GAMS/LINDOGlobal finalizó las 31 ejecuciones consumiendo el tiempo de computación disponible, como ya se ha mencionado anteriormente, mientras que prácticamente el resto de los algoritmos convergieron a una solución (o generaron un error) sin agotar el tiempo en al menos el 97 % de sus respectivas ejecuciones.

Algoritmo*	MILP [†]	NLP [‡]
GAMS/DICOPT	GAMS/BDMLP	GAMS/SNOPT
	GAMS/CBC	GAMS/SNOPT
	GAMS/CPLEX	GAMS/SNOPT
	GAMS/GUROBI	GAMS/SNOPT
	GAMS/XA	GAMS/SNOPT
	GAMS/XPRESS	GAMS/SNOPT

* Véase más información sobre los algoritmos en la tabla 3.1.

[†] Técnica auxiliar de programación lineal entera empleada. Véase más información en la tabla 3.2.

[‡] Técnica auxiliar de programación no lineal empleada. Véase más información en la tabla 3.3.

Tabla 3.6: Listado de algoritmos con los que se obtuvieron los mejores resultados en la experiencia numérica de la subsección 3.5.2. En concreto, se detallan los 6 algoritmos, todos variantes de GAMS/DICOPT, que devolvieron la mejor solución conocida para todos los ejemplos de la batería excepto los supuestamente infactibles.

De todo lo dicho se desprende que prácticamente todas las $8 \times 7 = 56$ variantes de GAMS/DICOPT, a excepción de las 8 en que se usó GAMS/MSNLP, alcanzaron en la gran mayoría de los ejemplos de la batería los mejores resultados; y en particular, las variantes detalladas en la tabla 3.6 devolvieron en todos los casos la mejor solución conocida.

Versión interactiva de los resultados En la página web http://personales.upv.es/vigibos/docs/phd_thesis/tables.html pueden consultarse sendas versiones interactivas de las tablas A.1 y A.2, en las que, por ejemplo, es posible ordenar los diferentes problemas y algoritmos, respectivamente, en orden creciente o decreciente respecto de cada una de las columnas presentes.

Conclusión del análisis descriptivo Tras este primer análisis descriptivo, en el apartado siguiente realizamos un estudio estadístico comparativo completo de los resultados de esta experiencia numérica, con el objetivo de

determinar qué problemas y, sobre todo, qué algoritmos (o combinaciones de ambos) permiten obtener mejores resultados.

Numéricamente es claro que los algoritmos listados en la tabla 3.6 son los que ofrecen mejores resultados. Aun así, resulta de interés determinar mediante técnicas estadísticas cuáles de las disparidades que se observan entre los resultados ofrecidos por los diferentes algoritmos son significativas y cuáles son, por el contrario, debidas al azar.

Análisis de los resultados

El objetivo de este apartado es determinar si alguno o algunos de los algoritmos probados ofrecen un comportamiento significativamente mejor que el resto, cuando son evaluados sobre problemas de OPC de los cuales la batería de ejemplos definida es representativa. Asimismo, también se contrastará si las condiciones iniciales del problema influyen en el comportamiento de los algoritmos.

Parámetros para evaluar el comportamiento de un algoritmo Las medidas más habituales (complementarias y no excluyentes) para evaluar el comportamiento de un algoritmo de optimización sobre un problema dado suelen ser las dos siguientes:

- el grado de desviación de la solución proporcionada por el algoritmo respecto de la mejor solución conocida para el problema (una medida de la *calidad* de la solución dada por el algoritmo, y por tanto de la *eficacia* de éste); y
- el tiempo que emplea el algoritmo en obtener dicha solución (una medida de su *eficiencia*).

En la presente experiencia, la duración media de las ejecuciones de los algoritmos en GAMS fue de 19,55 segundos, con un máximo de 1000 segundos (límite establecido por defecto), siendo los cuartiles primero y tercero 0,48 y 3,60 segundos, respectivamente. También en el caso de la herramienta Solver de Excel, cuyos tiempos no pudieron ser

recogidos¹², las ejecuciones se realizaron de manera casi instantánea. Los tiempos registrados son perfectamente asumibles en todos los casos y, por tanto, en esta ocasión concreta dicho parámetro no parece demasiado relevante para determinar la menor o mayor bondad de uno u otro algoritmo.

En consecuencia, utilizaremos únicamente la desviación respecto a la mejor solución conocida como principal medida del comportamiento de un algoritmo sobre un determinado modelo. Generalmente, dicha desviación se mide comparando en términos relativos (porcentaje) el valor de la función objetivo para la solución proporcionada por el algoritmo y el valor de dicha función asociado a la mejor solución encontrada para el problema que se está resolviendo. De manera más formal, si llamamos \mathcal{A} al conjunto de algoritmos tomados en consideración para resolver un determinado problema P , $s(P, A)$ a la mejor solución proporcionada por una ejecución (o, en ocasiones, un conjunto de ejecuciones) de un determinado algoritmo $A \in \mathcal{A}$ sobre P y $f(P, A)$ al valor de la función objetivo asociado a $s(P, A)$, definimos el mejor (en este caso, el menor) valor encontrado para el valor de la función objetivo del problema P mediante los algoritmos contenidos en \mathcal{A} como

$$f_{\text{best}}(P, \mathcal{A}) = \text{mín} \{ f(P, A) \mid A \in \mathcal{A} \} . \quad (3.37)$$

Podemos definir entonces la desviación de un algoritmo $A \in \mathcal{A}$ respecto de la mejor solución conocida para el problema P como

$$\text{dev}(P, A) = \left| \frac{f(P, A) - f_{\text{best}}(P, \mathcal{A})}{f_{\text{best}}(P, \mathcal{A})} \right| . \quad (3.38)$$

El principal inconveniente de esta manera de medir la desviación es que no está definida para el caso en que el denominador $f_{\text{best}}(P, \mathcal{A})$ (mínimo conocido para el valor de la función objetivo de P dentro de

¹²Hasta donde hemos podido averiguar, Excel no proporciona ningún modo de calcular el tiempo de CPU (tiempo efectivo de procesador) empleado por Solver en resolver un determinado problema.

su región factible) tome el valor 0. Esto en nuestro caso es imposible (es bastante claro, pero se evidenciará más aún en el capítulo 4), así que podemos considerarla como válida para nuestro propósito.

Otro detalle que debe resolverse es cómo definir $\text{dev}(P, A)$ en el caso en que el algoritmo A no proporcione ninguna solución para P (cosa que, como se ha visto, sucede en algunas ocasiones en nuestra experiencia numérica). Obviamente, si P tiene región factible no nula y A no es capaz de devolver ninguna solución para P , entonces la evaluación de A debería ser la peor posible. Formalmente, podríamos definir $\text{dev}(P, A)$ en ese caso como $+\infty$. De modo práctico, existen distintas alternativas válidas respecto a qué valor dar a $\text{dev}(P, A)$:

- Asignar a A la mayor desviación encontrada al resolver P mediante todos los algoritmos de \mathcal{A} , multiplicada por un determinado factor $K > 1$, esto es:

$$\text{dev}(P, A) = K \cdot \text{máx} \{ \text{dev}(P, B) \mid B \in \mathcal{A}, B \text{ devuelve solución} \}, K > 1. \quad (3.39)$$

- Asignarle la desviación media observada en \mathcal{A} multiplicada por un factor K bastante mayor que 1 (al menos 2, por ejemplo):

$$\text{dev}(P, A) = K \cdot \text{media} \{ \text{dev}(P, B) \mid B \in \mathcal{A}, B \text{ devuelve solución} \}, K \geq 2. \quad (3.40)$$

Tras realizar algunas pruebas, hemos decidido elegir el primer modo de asignar un valor a la desviación $\text{dev}(P, A)$ cuando A no proporciona solución para P , siempre que haya algún algoritmo en \mathcal{A} que sí lo haga; tomamos $K = 1,5$.

Únicamente resta especificar cómo tratar aquellos problemas para los cuales ninguno de los algoritmos considerados ha sido capaz de proporcionar ninguna solución factible; es el caso de los ejemplos 1 a 5 de nuestra batería, como ya se ha mencionado. Dichos ejemplos, todos con $C_p = 1,2$, pueden suponerse no factibles (si bien las técnicas que hemos empleado hasta el momento no permiten asegurar inequívocamente que lo sean), y necesariamente serán *descartados* para el análisis.

Esto nos deja con $36 - 5 = 31$ ejemplos en los que basar el análisis, entre ellos 4 problemas con $C_p = 1,2$, categoría que ahora queda en

cierto modo *incompleta* con respecto al resto. Dado que, además, el análisis numérico realizado en el apartado anterior sugiere que estos 4 ejemplos factibles con $C_p = 1,2$ presentan un comportamiento diferente a los demás (probablemente porque contienen muy pocas soluciones factibles), consideramos que es más conveniente analizarlos a parte. Es decir, realizaremos el análisis principal sobre la base de los $27 = 31 - 4$ ejemplos restantes (con $C_p \in \{1,3333; 1,4; 1,5\}$).

Calibración y comparación de algoritmos mediante análisis de la varianza Para llegar a determinar cuál o cuáles de las 135 combinaciones de algoritmos probadas en esta experiencia ofrecen resultados significativamente mejores en términos de la desviación, tal y como ha sido definida, cuando son aplicados sobre la batería de 27 ejemplos (que representan la clase de problemas que deseamos resolver), utilizaremos herramientas de inferencia estadística apropiadas. En concreto, se hará uso de un enfoque basado en diseño de experimentos (*design of experiments*; DOE) y análisis de la varianza (*analysis of variance*; ANOVA) (Fisher, 1925, 1935).

El ANOVA es un conjunto de técnicas de inferencia estadística que permite determinar y cuantificar la contribución de uno o varios factores (o, en general, *fuentes de variabilidad*), y sus posibles interacciones, sobre el comportamiento (concretamente, la variabilidad) de una determinada cantidad aleatoria continua, referida como variable dependiente. Por tanto, se ajusta al objetivo que se pretende en este apartado: averiguar si la elección de uno u otro tipo de algoritmo para la resolución de problemas de OPC tiene alguna influencia significativa en la calidad de las soluciones obtenidas, medida a través de la expresión (3.38)¹³. Dentro del ANOVA podemos distinguir los métodos clásicos de ANOVA paramétrico (que asumen que la variable de interés sigue una distribución normal para cada combinación de factores, entre otras hipótesis), y otros

¹³Podemos considerar la calidad de la solución resultante de ejecutar una determinada variante de un algoritmo como una variable aleatoria, en tanto en cuanto el resultado depende del problema sobre el cual se aplique el algoritmo y dicho problema es, en cierto modo, un elemento escogido aleatoriamente de una determinada clase o población (en nuestro caso, la de los problemas OPC).

métodos alternativos, llamados no paramétricos, que no presuponen ninguna estructura en los datos a analizar. A algunos de estos últimos se les achaca, en términos generales, una menor potencia para detectar diferencias que los correspondientes métodos paramétricos, mientras que estudios relativamente recientes demuestran que el ANOVA paramétrico es bastante robusto ante el incumplimiento de sus hipótesis y ofrece resultados al menos tan buenos como los test no paramétricos incluso en situaciones de extrema falta de normalidad (véanse por ejemplo [Rasch y Guiard, 2004](#) y [Basso et al., 2007](#)). Por estos motivos, se ha elegido utilizar la técnica clásica paramétrica de ANOVA para analizar los datos producidos por la presente experiencia numérica¹⁴. Algunos textos clásicos para profundizar en las técnicas relacionadas con el DOE y el ANOVA son los de [Ruiz-Maya Pérez \(1983\)](#), [Montgomery \(2013\)](#) y [Navidi \(2011\)](#). Acerca del uso del DOE y el ANOVA para la calibración de algoritmos puede consultarse el trabajo de [Ridge y Kudenko \(2010\)](#).

Estrategia y herramientas de análisis Del total de 135 combinaciones de algoritmos probadas, la mayor parte corresponden a variantes de los algoritmos GAMS/AlphaECP y GAMS/DICOPT (56 combinaciones cada uno), seguidos de GAMS/OPQNLP y GAMS/SBB (7 cada uno), resultantes, en cada caso, de considerar (y combinar, en su caso) las diferentes alternativas disponibles para los métodos auxiliares de NLP (y de MILP, en los dos primeros). Los 9 algoritmos restantes se presentan en una única versión (véase la tabla 3.4).

Procederemos, en primer lugar, a analizar cuál de las combinaciones de cada uno de los cuatro algoritmos recién mencionados proporciona mejores resultados. De ese modo, determinaremos la *mejor versión* de cada algoritmo, la que tomaremos como *representante* de cada uno de ellos. Así,

¹⁴En definitiva, la intención es hacer uso del ANOVA como herramienta válida para añadir una capa más de *certidumbre* a lo que se observe directamente a nivel muestral (siempre considerando que el conjunto de 27 problemas constituye una muestra aleatoria y representativa del total de problemas de OPC que pueda resultar de interés resolver). Queda fuera de los objetivos de la presente memoria un análisis más completo y, necesariamente, más complejo.

el número total de algoritmos a considerar quedará reducido a $4 + 9 = 13$, cuyas diferencias serán analizadas en una segunda etapa. En ambas etapas añadiremos también como factores del análisis los parámetros del problema C_p , α^* y β^* , con el fin de determinar si su valor se traduce en una menor o mayor dificultad para alcanzar la mejor solución. En términos de DOE, en la primera etapa estaremos analizando datos de un diseño con cinco (en GAMS/AlphaECP y GAMS/DICOPT) o cuatro (en el caso de GAMS/OPQNL y GAMS/SBB) factores cruzados de efectos fijos para cada algoritmo; en la segunda etapa trataremos con un diseño de cuatro factores cruzados de efectos fijos¹⁵.

Utilizaremos el contraste clásico basado en la distribución asintótica del estadístico F para evaluar la significatividad de cada factor y de las interacciones. Asimismo, se calcularán y graficarán los valores medios dentro de cada nivel y los respectivos intervalos basados en el método de la diferencia honestamente significativa de Tukey (1949) (*honestly significant difference*; intervalos HSD) para determinar, si es el caso, qué niveles de un determinado factor producen desviaciones significativamente distintas al resto, así como el sentido de las posibles interacciones. En todos los casos se trabajará con un nivel de confianza del 95 %.

Calibración del algoritmo GAMS/AlphaECP Comenzaremos analizando los resultados del algoritmo GAMS/AlphaECP. Para cada una de las 56 variantes de este algoritmo contamos con 27 observaciones, correspondientes a los resultados de ejecutar esa variante sobre cada uno de los 27 problemas tomados en consideración, transformados después en desviaciones según la expresión (3.38). Nótese que *la misma* colección de problemas es abordada con cada una de las 56 variantes. Si esos 27 pro-

¹⁵Los factores C_p , α^* y β^* podrían considerarse en cierto modo aleatorios, en el sentido de que los valores elegidos para cada uno de ellos en la presente experiencia numérica son una mera muestra supuestamente representativa de los posibles valores que pueden tomar. Sin embargo, en DOE este tipo de factores, cuyos niveles han sido fijados de antemano a valores relevantes en algún sentido y suficientemente diferentes entre sí, suelen considerarse de efectos fijos. En cualquier caso, los análisis que se mostrarán aquí se replicaron también considerando dichos factores como aleatorios, y los resultados y conclusiones que se obtuvieron fueron similares.

blemas hubieran sido generados de manera totalmente aleatoria y sin ninguna estructura, el método adecuado para analizar los datos sería, por tanto, lo que se conoce como un ANOVA de medidas repetidas (*repeated measures ANOVA*; rANOVA), para lo cual existen técnicas específicas.

Un enfoque alternativo y totalmente equivalente, que permite utilizar de forma directa el método clásico de ANOVA (el cual está diseñado suponiendo que las observaciones entre las diferentes combinaciones de los factores son independientes), consiste en tratar el conjunto de individuos o *unidades experimentales* de la muestra (en este caso, los 27 problemas) como un factor más del análisis (*factor de bloqueo*, utilizando terminología de DOE).

Además, en nuestro caso los 27 ejemplos *sí* presentan una estructura concreta, pues provienen de combinar los valores considerados para los tres parámetros del problema, C_p , α^* y β^* , cada uno con 3 niveles. Por ello, en lugar de introducir las unidades experimentales directamente como un único factor, trataremos como factores del diseño los tres parámetros citados, como ya habíamos avanzado más arriba.

De esta manera, al considerar C_p , α^* y β^* como factores de bloqueo, lo que se consigue, por una parte, es poder estimar mejor mediante el ANOVA la influencia de los factores objeto de interés (los algoritmos auxiliares de MILP y NLP) en la calidad de la solución proporcionada por GAMS/AlphaECP al problema del OPC, ya que se *desgaja* de la varianza residual la contribución de los parámetros del problema; y, por otro lado, se podrá conocer si el algoritmo se comporta mejor o peor en función de cuáles sean los valores de dichos parámetros, y si hay algún tipo de interacción entre éstos y los métodos de MILP y NLP.

Así, aplicaremos un ANOVA con cinco factores cruzados (el método de MILP, el método de NLP, y los parámetros C_p , α^* y β^*), tomando como variable dependiente la desviación relativa respecto de la mejor solución (3.38) y considerando la expresión (3.40) como forma de asignar valores en aquellos casos en que una versión concreta del algoritmo no devuelva una solución para un determinado problema; tomaremos en cuenta todas las posibles interacciones de segundo orden. Nótese que

estamos considerando un total de $8 \times 7 \times 3 \times 3 \times 3 = 1512$ combinaciones, con una observación para cada una de ellas (lo que se conoce como ANOVA sin repetición).

Sólo en 5 (menos de un 1%) de las 1512 experiencias el algoritmo GAMS/AlphaECP no obtuvo ninguna solución. En esos casos, como ya se ha dicho, se completó el dato de la desviación usando la mayor desviación observada para el problema correspondiente multiplicada por 1,5. Fue el caso del ejemplo 13 ($C_p = 1,3333$, $\alpha^* = 0,005$, $\beta^* = 0,005$) con GAMS/GUROBI y GAMS/CONOPT como algoritmos auxiliares de MILP y NLP, respectivamente; y el caso del ejemplo 34 ($C_p = 1,5$, $\alpha^* = 0,01$, $\beta^* = 0,005$) con GAMS/CPLEX como algoritmo de MILP y GAMS/CONOPT, GAMS/IPOPT, GAMS/PATHNLP y GAMS/SNOPT como algoritmos de NLP.

En las restantes $1512 - 5 = 1507$ experiencias con GAMS/AlphaECP la desviación media observada fue de 10,76%, siendo los cuartiles primero y tercero 0,00% y 13,12%, respectivamente.

Tanto este como el resto de análisis de la varianza llevados a cabo se realizaron con la herramienta Statgraphics (versión 16.2.04 para ordenadores con sistema operativo Windows y procesador a 64 bits). La salida completa del ANOVA y las medias calculadas se muestran en las tablas A.3 a A.6 del anexo A.

Según el análisis realizado, la desviación media global observada al resolver la batería de 27 problemas con GAMS/AlphaECP (tras asignar valores a los casos en que no se produjo solución) es de 11,14%. De los dos factores objeto de investigación, relacionados con el algoritmo, el único con efectos significativos es el método de MILP (p -valor $< 0,0001$; 0,0904 para el método de NLP). También se detecta que los tres parámetros del problema, C_p , α^* , β^* , tienen influencia en la calidad de la solución final (p -valor $< 0,0001$ en los tres casos). No se aprecian efectos significativos debidos a interacciones entre el método de MILP y el de NLP (p -valor $> 0,9999$; figura 3.3(a)). En cuanto a las interacciones de orden 2 de los métodos de MILP y NLP con los parámetros del problema, son significativas las tres que involucran al método de MILP (p -valor

$< 0,0001$ en los tres casos). Por último, también son significativas las tres interacciones de segundo orden resultantes de combinar los parámetros del problema (p -valores igual a $0,0001$, menor que $0,0001$ e igual a $0,0014$ para $C_p \cdot \alpha^*$, $C_p \cdot \beta^*$ y $\alpha^* \cdot \beta^*$, respectivamente).

De acuerdo con el diagrama de medias e intervalos HSD correspondientes al método de MILP (figura 3.2(a)), los tres algoritmos que han ofrecido desviaciones significativamente menores que el resto son GAMS/GUROBI, GAMS/SULUM y GAMS/XA; entre ellos parece no haber diferencias significativas, si bien GAMS/SULUM presenta en la muestra una desviación media inferior en un 37% a la de GAMS/XA (2,25% frente a 3,56%, según la tabla A.4 del anexo A) y un 60% inferior a la de GAMS/GUROBI (5,65%).

Por su parte, el gráfico de medias e intervalos HSD recogido en la figura 3.2(b) muestra que el método auxiliar de NLP empleado no contribuye significativamente a la variabilidad de la desviación, si bien se aprecia que el algoritmo GAMS/MSNLP presenta en la muestra el valor medio más bajo (9,23%, según la tabla A.4 del anexo A).

El gráfico de medias e intervalos HSD correspondiente al parámetro C_p (figura 3.2(c)) confirma que, en promedio, cuanto menos restrictivo es el valor del parámetro (es decir, cuanto mayor es), más fácil resulta para las diferentes variantes del algoritmo GAMS/AlphaECP encontrar la mejor solución conocida. Algo parecido sucede con el parámetro β^* , como se aprecia en la figura 3.2(e).

En cambio, con el parámetro α^* sucede a la inversa (figura 3.2(d)): cuanto mayor es su valor, peor es, aparentemente, la calidad de la solución encontrada, en promedio. La causa de esto puede estar en el hecho, ya mencionado, de que en 4 de las 5 ejecuciones en que GAMS/AlphaECP no devolvió solución el valor de α^* era 0,01. Al parecer, es el valor de la desviación asignado a esas 4 experiencias el que contribuye a aumentar la desviación media para dicho nivel del factor α^* .

Las interacciones significativas detectadas por el análisis parecen explicarse también, en parte, por las características de las 5 experiencias que no devolvieron solución. Así, por ejemplo, se observa un valor

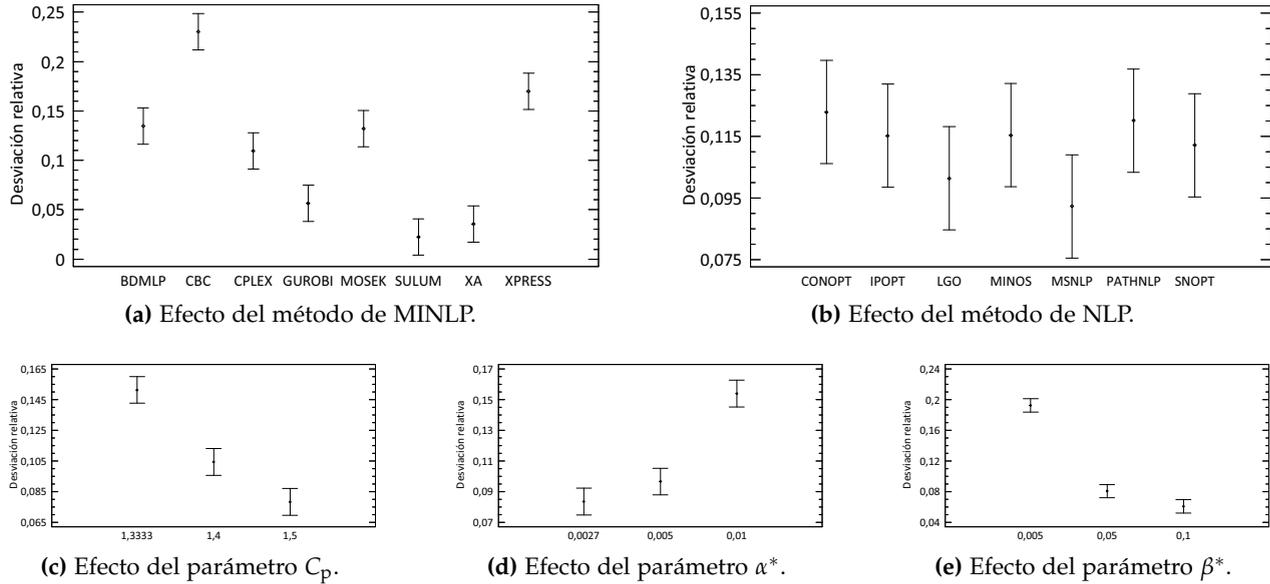
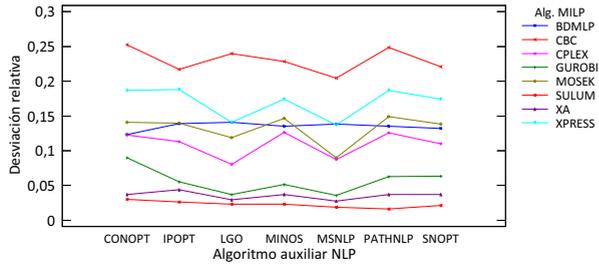
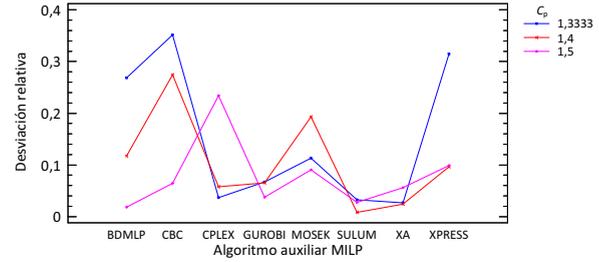


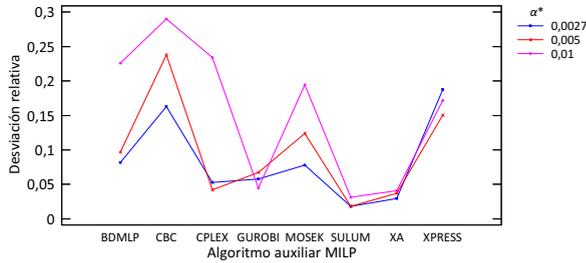
Figura 3.2: Gráficos de medias e intervalos HSD para los efectos principales del ANOVA correspondiente a la calibración del algoritmo GAMS/AlphaECP.



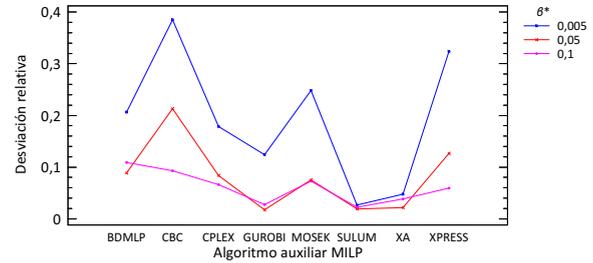
(a) Interacción MILP·NLP.



(b) Interacción MILP· C_p .

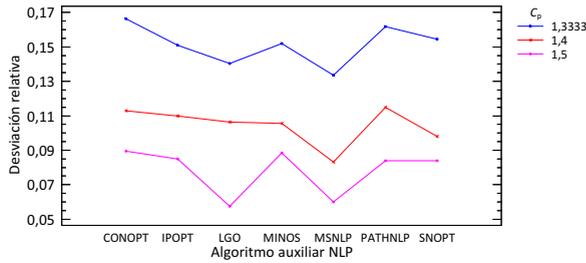


(c) Interacción MILP· α^* .

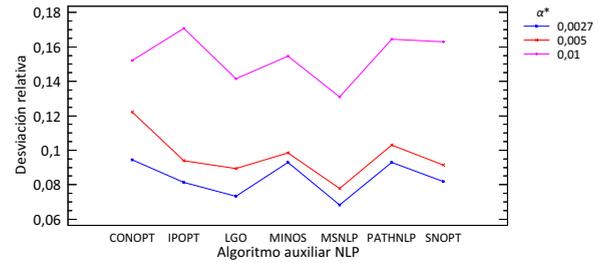


(d) Interacción MILP· β^* .

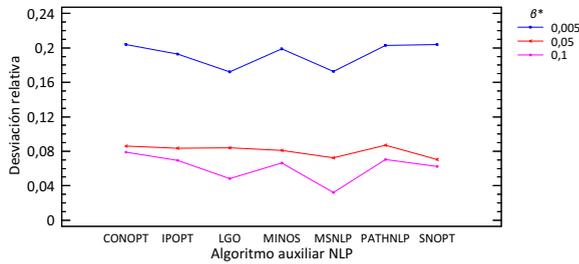
Figura 3.3: Gráficos de medias para las interacciones de segundo orden del ANOVA correspondiente a la calibración del algoritmo GAMS/AlphaECP. *Continúa en la página siguiente.*



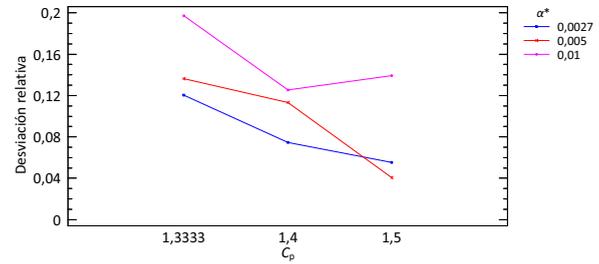
(e) Interacción NLP · C_p .



(f) Interacción NLP · α^* .



(g) Interacción NLP · β^* .



(h) Interacción C_p · α^* .

Figura 3.3: Gráficos de medias para las interacciones de segundo orden del ANOVA correspondiente a la calibración del algoritmo GAMS/AlphaECP. *Continúa en la página siguiente.*

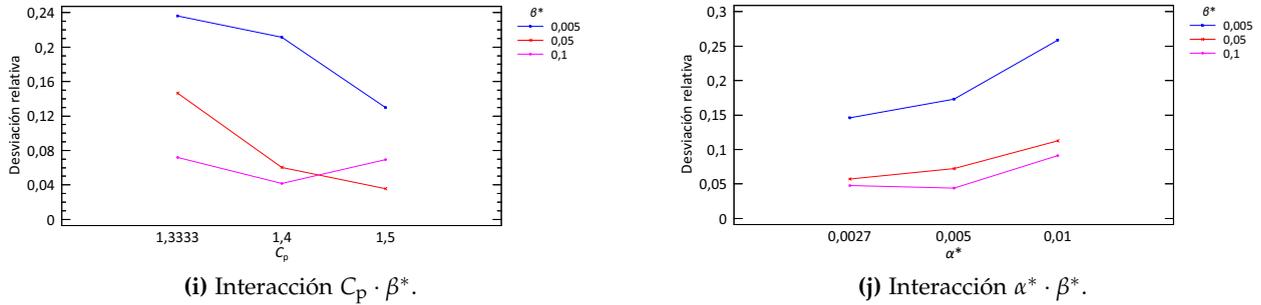


Figura 3.3: Gráficos de medias para las interacciones de segundo orden del ANOVA correspondiente a la calibración del algoritmo GAMS/AlphaECP.

especialmente elevado de la desviación media asociada a la combinación del algoritmo auxiliar de MILP GAMS/CPLEX junto con el valor $C_p = 1,5$, que se corresponde con la configuración de 4 de las 5 ejecuciones sin solución (figura 3.3(b)). De igual forma, la combinación $C_p = 1,5$ con $\alpha^* = 0,01$ (figura 3.3(h)), que se da en esas mismas 4 experiencias, conlleva una desviación media visiblemente alta.

En conclusión, dado que los contrastes sobre el método de NLP y sobre las interacciones no revelan diferencias significativas, una recomendación genérica, en caso de emplear el algoritmo GAMS/AlphaECP para resolver problemas de OPC, sería utilizar GAMS/SULUM (o de modo alternativo GAMS/XA o GAMS/GUROBI) como métodos auxiliares de MILP, independientemente del método de NLP.

Por otro lado, analizando la tabla de medias A.6, puede concluirse que la configuración óptima del algoritmo GAMS/AlphaECP cuando se usa para resolver problemas de OPC, que será la que tomemos como *representante* del algoritmo en la segunda fase de nuestro análisis, consiste en utilizar simultáneamente GAMS/SULUM como método de MILP y GAMS/PATHNLP como método de NLP, lo cual abreviaremos como GAMS/AlphaECP(sulum,pathnlp). Esta combinación concreta produjo en los experimentos realizados la menor desviación promedio (1,61 %); concretamente en 16 de los 27 ejemplos consiguió alcanzar la mejor solución conocida, y en el resto la desviación media que ofreció fue 3,95 %. La siguiente configuración con menor desviación promedio (1,84 %) fue GAMS/AlphaECP(sulum,msnlp), la cual alcanzó la mejor solución conocida en 15 de los 27 problemas.

Tanto en este caso como en los posteriores, podríamos haber dado una recomendación particular respecto a qué métodos auxiliares de MILP y NLP elegir en función de cuáles sean los valores de los parámetros C_p , α^* y β^* , en lugar de una configuración *ganadora en promedio* para los 27 problemas. Dicha recomendación consistiría en elegir una combinación mejor que GAMS/AlphaECP(sulum,pathnlp), si la hay, en los $27 - 16 = 11$ problemas en los que dicha configuración no alcanza la mejor solución conocida, y estaría basada en los valores de desviación observados para

cada combinación concreta de los factores, o bien en la estimación de parámetros del modelo lineal subyacente resultante del ANOVA. Sin embargo, creemos que esto nos aleja de uno de los objetivos que hay detrás de esta experiencia: dar recomendaciones que un usuario no avanzado pueda seguir fácilmente.

Calibración del algoritmo GAMS/DICOPT El enfoque tomado con GAMS/DICOPT fue, en principio, idéntico al de la calibración del algoritmo GAMS/AlphaECP, ya que ambos admiten 8 posibles algoritmos auxiliares de MILP y 7 de NLP.

En el caso de GAMS/DICOPT, un total de 239 (15,8 %) de las 1512 experiencias no devolvieron ninguna solución. El método auxiliar de NLP empleado en el 87,03 % de este grupo de experiencias fue GAMS/MSNLP, seguido de GAMS/MINOS con un 7,95 %. De las 8 alternativas para el método auxiliar de MILP, destaca en esas mismas 239 experiencias sin solución una presencia ligeramente elevada de GAMS/SULUM, seguido de GAMS/MOSEK (un 15,90 % y un 13,39 %, respectivamente, frente a un $1/8 = 12,50$ % que sería lo esperable si hubiera independencia). Todo ello se ilustra mediante las figuras 3.4(a) y 3.4(b).

Entre aquellas experiencias que sí devolvieron solución, únicamente 42 (un 3,3 %) no alcanzaron la mejor solución conocida para cada problema, siendo además la desviación media observada en este grupo de sólo 0,12 %.

Tras asignar los valores correspondientes a las 239 experiencias sin solución válida, la desviación media global es de 13,16 %. En este caso, el ANOVA tomando como factores los dos algoritmos auxiliares y los tres parámetros del problema *no* se pudo realizar, debido a la abundancia de desviaciones observadas con valor igual a cero. Por ello, en este caso se decidió considerar las 27 unidades experimentales como un único factor, tal y como ya se ha explicado en la calibración de GAMS/AlphaECP.

El ANOVA (tabla A.7 del anexo A) reveló que tanto el método de MILP empleado como el de NLP contribuyen significativamente a explicar la variabilidad observada en la desviación (ambos *p*-valores inferiores

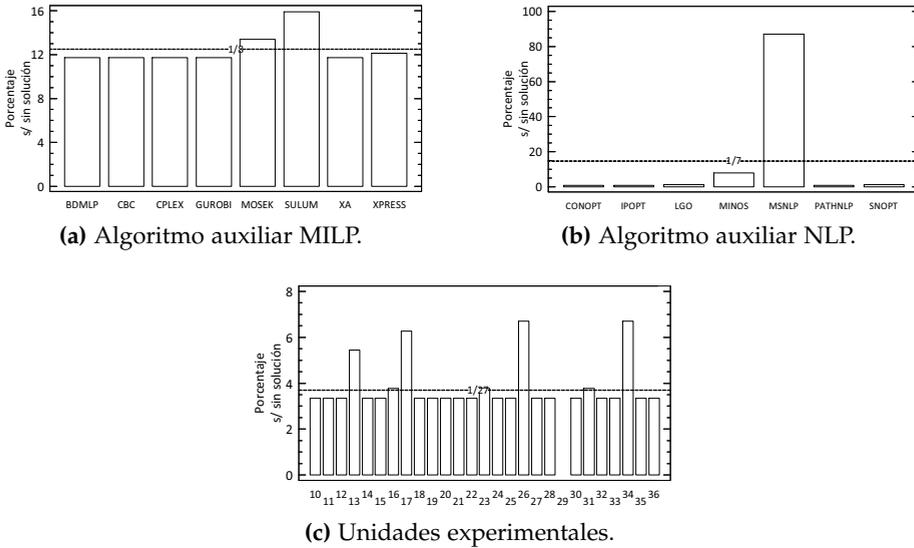


Figura 3.4: Distribución de los algoritmos auxiliares de MILP y NLP y de las unidades experimentales en las ejecuciones de GAMS/DICOPT que no devolvieron solución.

a 0,0001). Con grados de libertad similares, el estadístico F para los respectivos contrastes es mucho mayor para el segundo factor que para el primero (4114,87 frente a 13,04), lo cual se interpreta como que la elección del algoritmo auxiliar de NLP es la que más influye en la calidad de la solución. La interacción entre ambos métodos no resultó significativa (p -valor = 0,5288).

El análisis descriptivo realizado arriba, junto con el de los gráficos de medias correspondientes a ambos factores (figuras 3.5(a) y 3.5(b)) sugieren que, en realidad, lo que está influyendo en la calidad de la solución proporcionada por las diferentes variantes de GAMS/DICOPT es el hecho en sí que de que el algoritmo proporcione solución o no, ya que, según se ha visto, en caso de proporcionarla ésta es casi siempre la mejor conocida. Así, se observan *picos* en los gráficos de las desviaciones medias (tras asignar valores a las experiencias sin solución) asociados a los algoritmos que mayor proporción de ejecuciones sin solución presentan:

los ya mencionados GAMS/SULUM y GAMS/MOSEK en el caso del método de MILP, y GAMS/MSNLP y GAMS/MINOS, para el factor NLP.

Para contrastar esta hipótesis analizaremos directamente mediante un test no paramétrico exacto la relación entre el hecho de que el algoritmo termine con o sin solución y el método auxiliar de MILP o NLP empleado. Dado que la interacción entre ambas clases de métodos parece no ser significativa en esta ocasión, analizaremos ambos casos de manera separada.

Concretamente, se ejecutaron sendos tests exactos basados en permutaciones tomando como referencia el estadístico ji-cuadrado de Pearson (Agresti, 2002); se utilizó para ello el paquete informático StatXact 10 (Mehta, 1991). Más información sobre contrastes exactos basados en permutaciones (campo también conocido como *inferencia permutacional*) puede encontrarse en Good (1993), Edgington y Onghena (2007) y Cytel Software Corporation (2013).

Tanto en el caso del método auxiliar de MILP como en el de NLP el test confirmó que existen diferencias significativas en la proporción de ejecuciones que terminan sin solución en función del método utilizado (p -valores de 0,008526 y $1,495 \cdot 10^{-216}$, respectivamente).

El ANOVA detectó también que el problema a resolver influye en la desviación media obtenida (p -valor $< 0,0001$; figura 3.5(c)) e interacciona tanto con el método de MILP como con el de NLP (ambos p -valores inferiores a 0,0001; figuras 3.6(b) y 3.6(c)).

Como se puede deducir contrastando el gráfico de las desviaciones medias para cada problema (figura 3.5(c)) con el gráfico de distribución de las ejecuciones sin solución por unidad experimental (figura 3.4(c)), lo que realmente sucede, de nuevo, es que el algoritmo GAMS/DICOPT, en sus diferentes variantes, encuentra más dificultades para hallar solución en determinados problemas de la batería. Esa ausencia de solución se traduce en un incremento de la desviación media para ese problema, por la manera en como se ha decidido asignar valores en dichos casos.

En conclusión, la recomendación genérica sería evitar configurar GAMS/DICOPT con los métodos de MILP GAMS/SULUM y

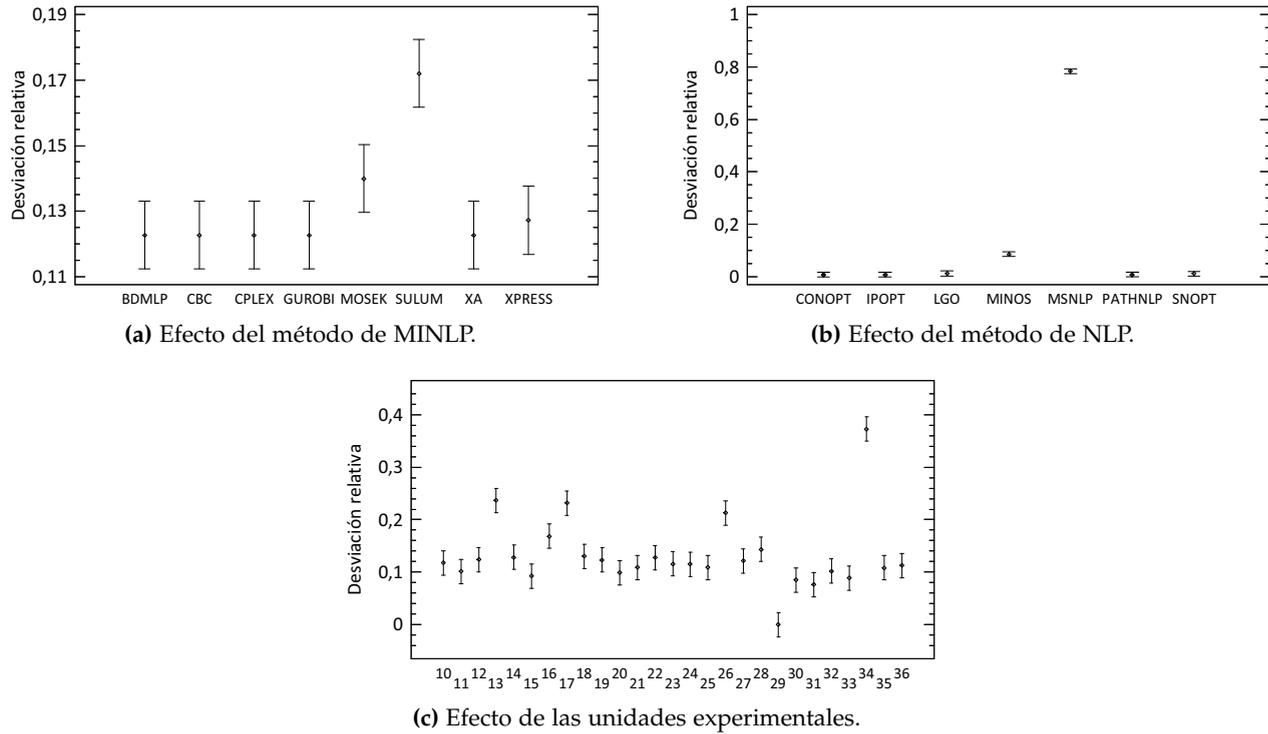
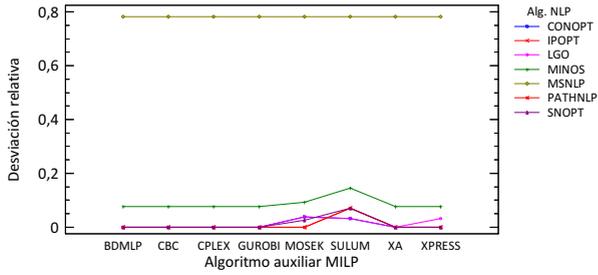
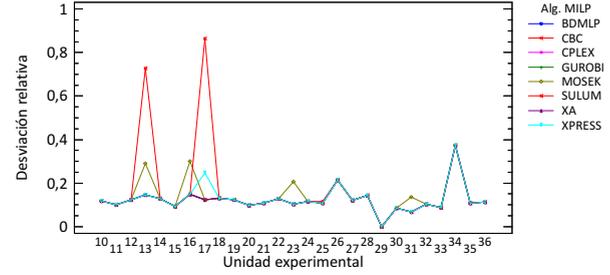


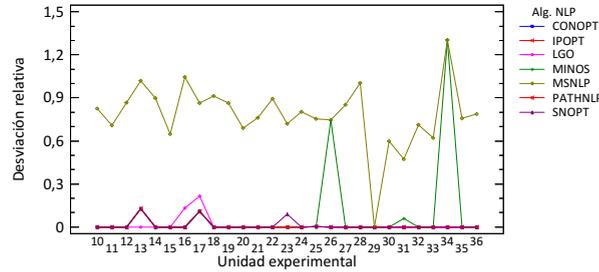
Figura 3.5: Gráficos de medias e intervalos HSD para los efectos principales del ANOVA correspondiente a la calibración del algoritmo GAMS/DICOPT.



(a) Interacción MILP·NLP.



(b) Interacción MILP· unidades experimentales.



(c) Interacción NLP· unidades experimentales.

Figura 3.6: Gráficos de medias para las interacciones de segundo orden del ANOVA correspondiente a la calibración del algoritmo GAMS/DICOPT.

GAMS/MOSEK y los métodos de NLP GAMS/MSNLP y GAMS/MINOS, cuando se use para resolver problemas de OPC.

En concreto, basándonos en los valores observados en la muestra (tabla A.10 del anexo A), las combinaciones que ofrecen una menor desviación promedio son aquellas en que intervienen los 6 métodos de MILP no descartados arriba (es decir, GAMS/BDMLP, GAMS/CBC, GAMS/CPLEX, GAMS/GUROBI, GAMS/XA y GAMS/XPRESS) junto con GAMS/SNOPT como método de NLP, con las cuales se alcanza en los 27 casos la mejor solución conocida (es decir, proporcionan una desviación media de 0%), seguidas de otras que involucran los mismos métodos de MILP y los métodos de NLP GAMS/CONOPT, GAMS/IPOPT, GAMS/LGO y GAMS/PATHNLP (todos los restantes excepto los dos a evitar); estas últimas proporcionan desviaciones medias del orden de 10^{-5} %. Tomaremos el primer grupo como representante de GAMS/DICOPT para la segunda fase del análisis, si bien es cierto que con cualquiera de las otras combinaciones se obtendrían resultados parecidos. Denotaremos el grupo de variantes elegido como GAMS/DICOPT(bdmlp | cbc | cplex | gurobi | xa | xpress,snopt), y lo abreviaremos como GAMS(...,snopt) por simplicidad.

Calibración del algoritmo GAMS/OQNLP La estrategia seguida con GAMS/OQNLP es la misma que con los dos algoritmos anteriores, sólo que en el caso presente existe un único factor objeto de interés con 7 niveles, el algoritmo de NLP, al que se añaden como factores de bloqueo los parámetros del problema C_p , α^* y β^* , igual que en las calibraciones anteriores.

El número de experiencias realizadas en este caso es, por tanto, $7 \times 3 \times 3 \times 3 = 189$. En *todas* ellas el algoritmo devolvió una solución factible. La desviación promedio global fue de 19,82%, con 10,03% como cuartil primero y 27,83% como cuartil tercero.

El ANOVA (tabla A.11 del anexo A) determina que el método de NLP empleado para ejecutar GAMS/OQNLP sí influye significativamente en la calidad de la solución final (p -valor $< 0,0001$).

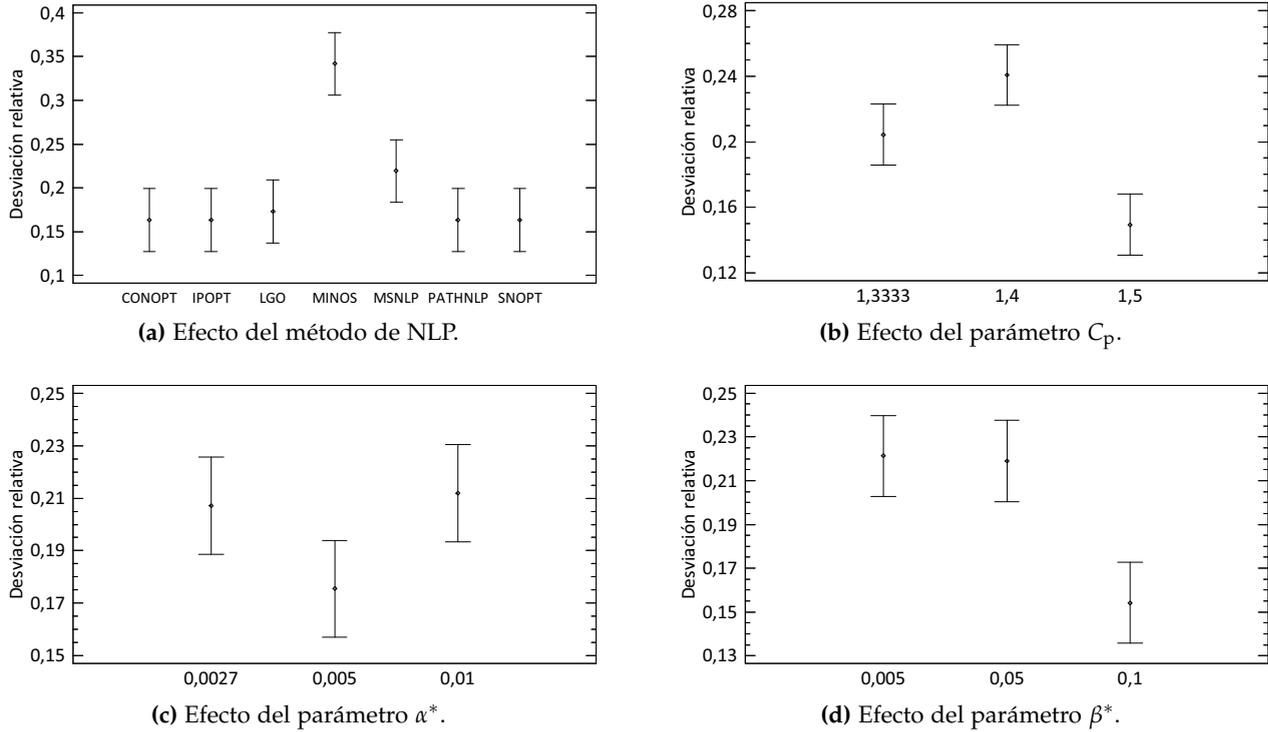


Figura 3.7: Gráficos de medias e intervalos HSD para los efectos principales del ANOVA correspondiente a la calibración del algoritmo GAMS/OQNLP.

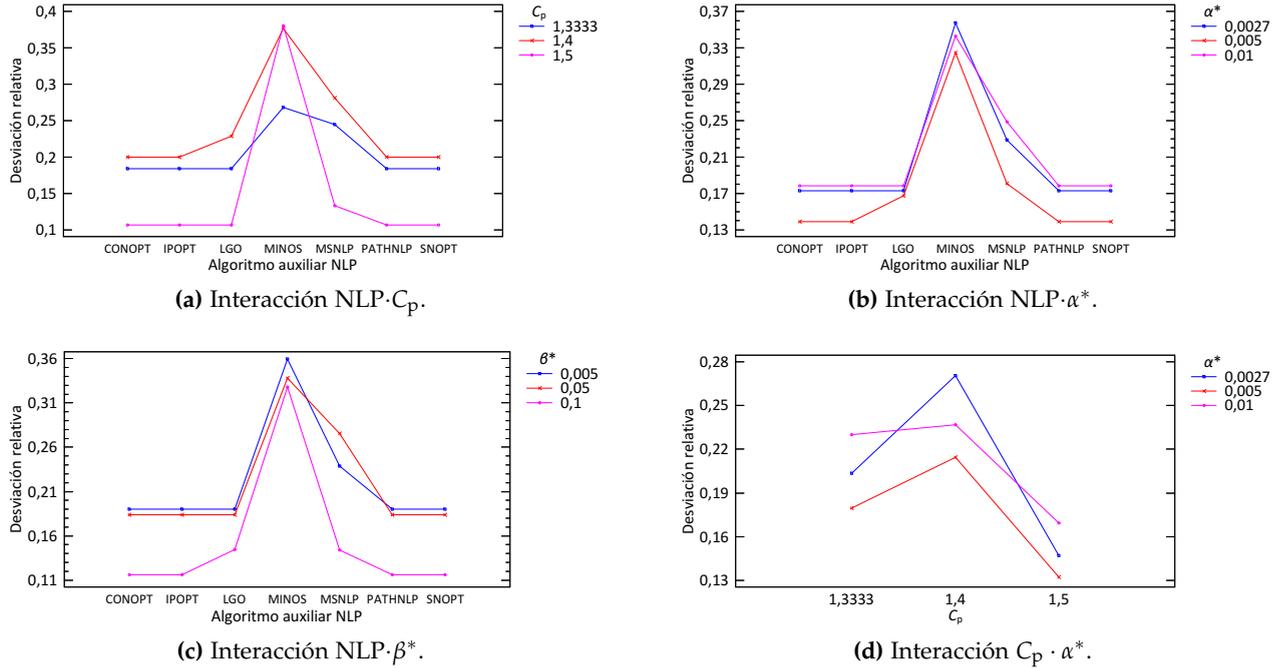


Figura 3.8: Gráficos de medias para las interacciones de segundo orden del ANOVA correspondiente a la calibración del algoritmo GAMS/OQNLP. *Continúa en la página siguiente.*

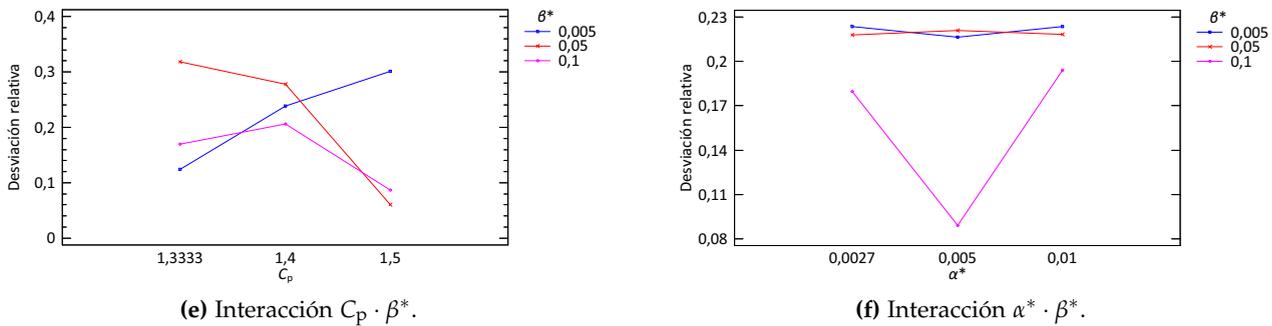


Figura 3.8: Gráficos de medias para las interacciones de segundo orden del ANOVA correspondiente a la calibración del algoritmo GAMS/OQNLP.

El gráfico de medias (figura 3.7(a)) revela que todos los métodos de NLP producen una desviación media similar, excepto GAMS/MINOS, con una desviación significativamente mayor. El método GAMS/MSNLP presenta también una desviación media bastante elevada, si bien la técnica de la HSD de Tukey no la considera significativamente distinta a las otras. Según la tabla A.12 del anexo A el método que produjo una desviación media menor fue GAMS/SNOPT (16,3306 %), prácticamente idéntica a la conseguida con GAMS/CONOPT, GAMS/IPOPT y GAMS/PATHNLP (16,3308 %).

Por otro lado, el ANOVA (tabla A.11 del anexo A) no detectó interacciones significativas entre el método de NLP y los parámetros C_p , α^* y β^* (p -valores de 0,0630, 0,9999 y 0,9551, respectivamente; figuras 3.8(a), 3.8(b) y 3.8(c)). Aun así, se observa (figura 3.8(a)) cómo el uso del método auxiliar GAMS/MINOS sí provoca un empeoramiento de la calidad media de las soluciones destacadamente mayor de lo esperado cuando $C_p = 1,5$.

En conclusión, la recomendación genérica si se usa GAMS/OQNLP para resolver problemas de OPC es evitar configurarlo con los métodos GAMS/MINOS y GAMS/MSNLP. Basándonos en los valores observados (tabla A.12 del anexo A), tomaremos GAMS/OQNLP(snopt) como variante del algoritmo a evaluar en la segunda fase del análisis, si bien con GAMS/CONOPT, GAMS/IPOPT y GAMS/PATHNLP se obtendrían resultados similares.

Calibración del algoritmo GAMS/SSB GAMS/SBB se presenta en las mismas 7 variantes que el algoritmo GAMS/OQNLP que acabamos de calibrar, y por tanto el procedimiento a seguir será el mismo.

En 10 (5,3 %) de las 189 ejecuciones de GAMS/SBB no se consiguió obtener ninguna solución; en todas ellas se empleó GAMS/MINOS como algoritmo auxiliar de NLP (problemas 13, 15, 19, 20, 21, 26, 27, 30, 32 y 34). En 138 de las ejecuciones restantes (un 77,1 % de ellas) se alcanzó la mejor solución conocida para el problema correspondiente. La desviación media entre las experiencias que devolvieron una solución distinta de la

mejor conocida fue 2,85 %.

Tras asignar valores a las ejecuciones sin solución, la desviación media global observada fue 4,95 %.

El ANOVA realizado (tabla A.14 del anexo A) no detecta diferencias significativas entre las desviaciones causadas por el método auxiliar de NLP utilizado (p -valor = 0,5906). Tampoco revela interacciones significativas entre éste y los parámetros C_p , α^* y β^* (p -valores de 0,6746, 0,7398 y 0,9055, respectivamente; figuras 3.10(a), 3.10(b) y 3.10(c)).

El gráfico de medias (figura 3.9(a)) confirma lo revelado por el ANOVA, si bien muestra una desviación media ligeramente mayor, lejos de ser significativamente distinta a las otras, en las ejecuciones configuradas con GAMS/PATHNLP, GAMS/MSNLP y GAMS/SNOPT respecto al resto. La recomendación general sería, por tanto, evitar preferiblemente estos tres métodos cuando se utilice GAMS/SBB como algoritmo para solucionar problemas de OPC. En particular, de acuerdo con la tabla de medias A.15 del anexo A, la configuración con menor desviación promedio en la muestra es GAMS/SBB(conopt) (3,082 %), y por ello será la que elijamos como representante de este algoritmo.

Comparación de la eficacia de los algoritmos Una vez hemos determinado en los apartados anteriores cuál es, de acuerdo con los resultados de la experiencia numérica, la mejor variante de los algoritmos con más de una versión (GAMS/AlphaECP, GAMS/DICOPT, GAMS/OQNLP y GAMS/SBB), pasamos a una segunda etapa en la que se compararán entre sí las eficacias de los 13 algoritmos base (los 4 ya calibrados más los 9 restantes; véase la tabla 3.7), considerados como herramientas de resolución de problemas de OPC. Al mismo tiempo, se analizará en qué grado las condiciones particulares del problema de OPC (concretadas en los parámetros C_p , α^* y β^*) contribuyen al menor o mayor éxito de los algoritmos en su búsqueda de la mejor solución.

De manera similar a los análisis ya realizados, tratamos las observaciones resultantes de la experiencia numérica como un diseño con cuatro factores fijos —el algoritmo, como factor objeto de interés, y los

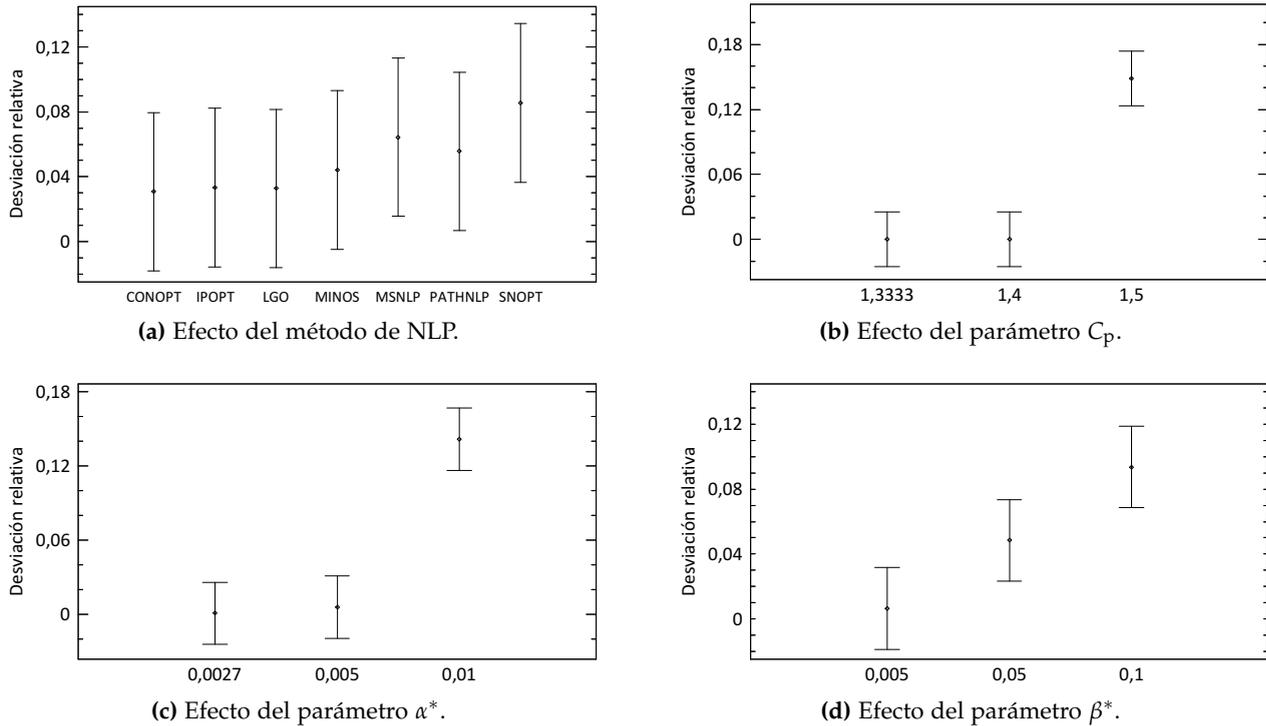
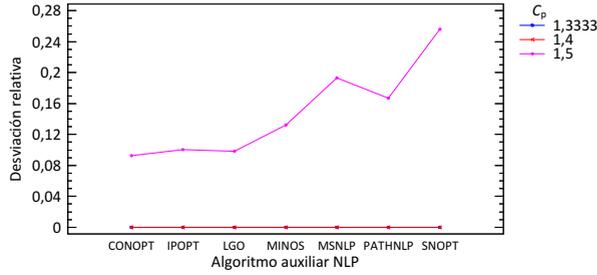
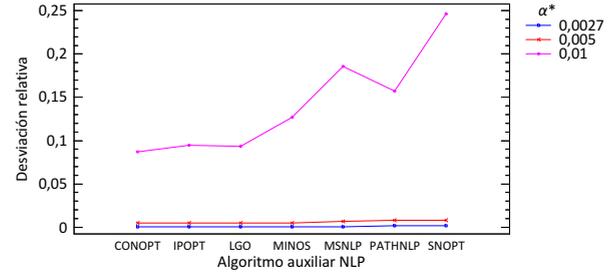


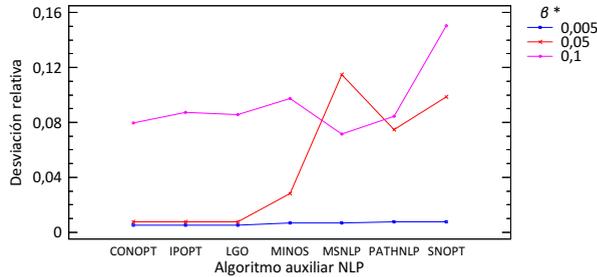
Figura 3.9: Gráficos de medias e intervalos HSD para los efectos principales del ANOVA correspondiente a la calibración del algoritmo GAMS/SBB.



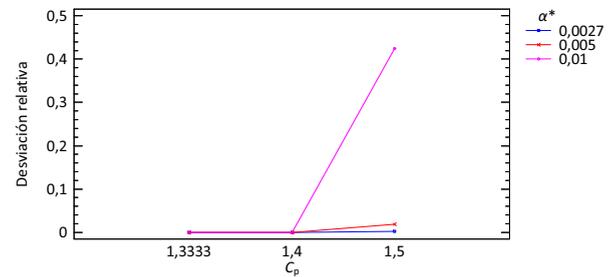
(a) Interacción NLP·Cp.



(b) Interacción NLP·α*.



(c) Interacción NLP·β*.



(d) Interacción Cp·α*.

Figura 3.10: Gráficos de medias para las interacciones de segundo orden del ANOVA correspondiente a la calibración del algoritmo GAMS/SBB. *Continúa en la página siguiente.*

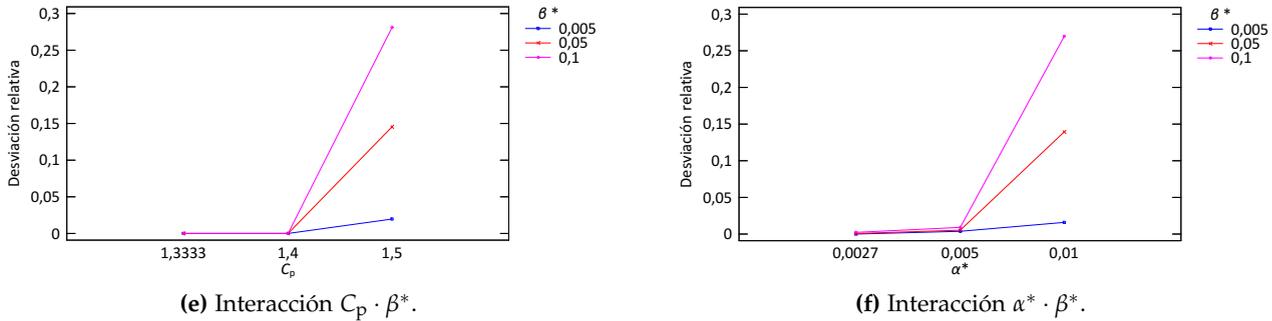


Figura 3.10: Gráficos de medias para las interacciones de segundo orden del ANOVA correspondiente a la calibración del algoritmo GAMS/SBB.

Algoritmo*	Algoritmo*
Excel Solver	GAMS/Bonmin B-iFP
GAMS/AlphaECP(sulum,pathnlp)	GAMS/DICOPT(...,snopt) [†]
GAMS/Bonmin B-BB	GAMS/KNITRO
GAMS/Bonmin B-ECP	GAMS/LINDOGlobal
GAMS/Bonmin B-OA	GAMS/OQNLP(snopt)
GAMS/Bonmin B-QG	GAMS/SBB(conopt)
GAMS/Bonmin B-Hyb	

—
^{*} Véase más información sobre los algoritmos en la tabla 3.1.
[†] En este caso, GAMS/DICOPT(...,snopt) sirve para representar de forma abreviada las variantes GAMS/DICOPT(bdmlp | cbc | cplex | gurobi | xa | xpress,snopt).

Tabla 3.7: Listado de algoritmos a comparar en la segunda etapa de la experiencia numérica de la subsección 3.5.2. Entre paréntesis se indica, en su caso, qué algoritmos auxiliares se utilizan.

parámetros C_p , α^* y β^* , como factores de bloqueo— y las analizamos utilizando ANOVA.

La tabla A.17 del anexo A muestra el resultado del ANOVA.

El ANOVA revela que el algoritmo utilizado influye significativamente en la variabilidad observada en la calidad de la solución (p -valor $< 0,0001$). El gráfico de medias e intervalos HSD (figura 3.11(a)) permite distinguir tres grandes grupos de algoritmos en cuanto a su eficacia. En primer lugar, existe un grupo, al que llamaremos *A*, formado por los algoritmos GAMS/AlphaECP(sulum,pathnlp), GAMS/Bonmin B-BB, GAMS/DICOPT(...,snopt), GAMS/KNITRO, GAMS/LINDOGlobal y GAMS/SBB(conopt), y que presenta los mejores resultados, con una desviación media global de 0,76 %. En segundo lugar, se distingue el *grupo B*, formado por los algoritmos GAMS/Bonmin B-OA, GAMS/Bonmin B-QG y GAMS/OQNLP(snopt), con resultados de calidad intermedia (desviación promedio global de 15,31 %). El resto de algoritmos (Excel Solver, GAMS/Bonmin B-ECP, GAMS/Bonmin B-Hyb y GAMS/Bonmin B-iFP) conforman el *grupo C*, con la desviación promedio global más elevada

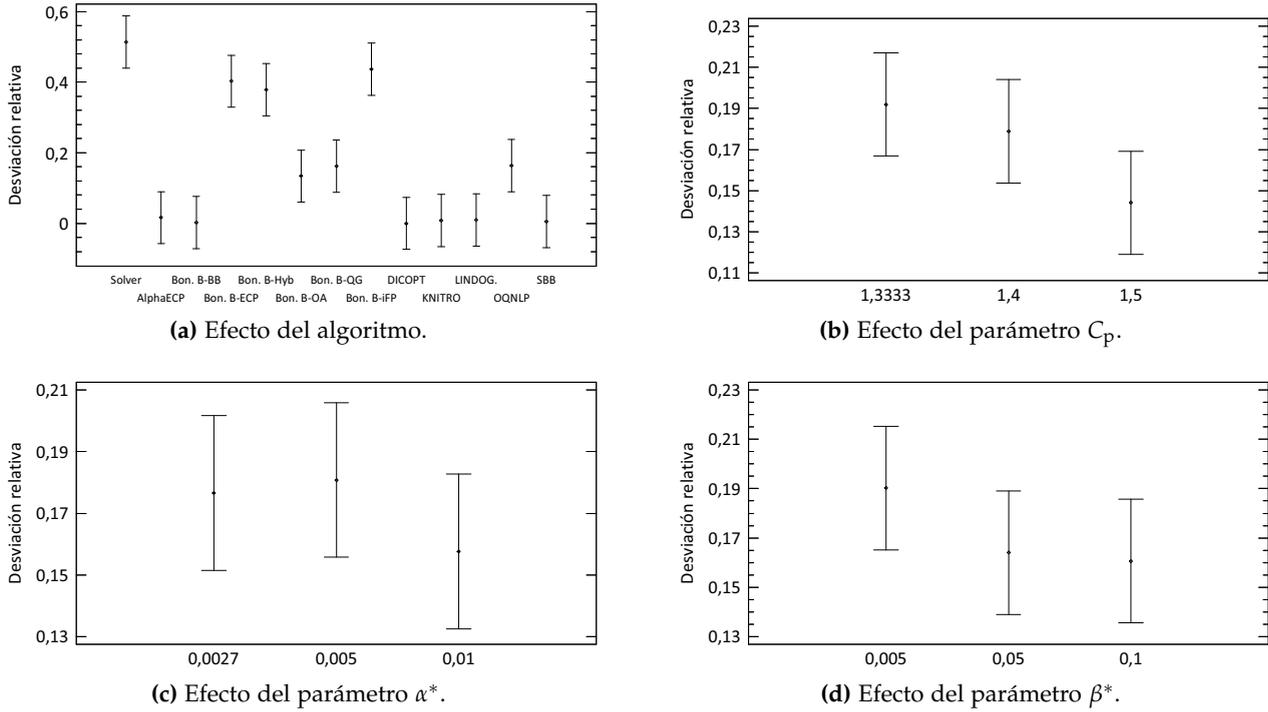


Figura 3.11: Gráficos de medias e intervalos HSD para los efectos principales del ANOVA correspondiente a la comparación de los algoritmos considerados en la experiencia numérica de la subsección 3.5.2.

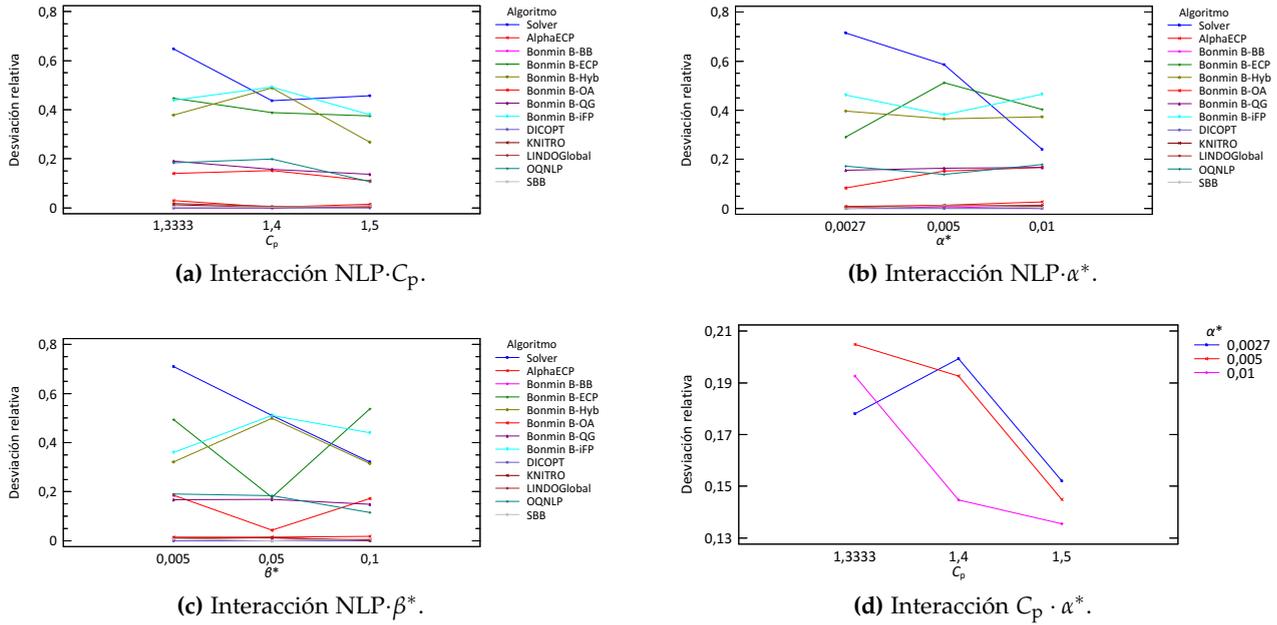


Figura 3.12: Gráficos de medias para las interacciones de segundo orden del ANOVA correspondiente a la comparación de los algoritmos considerados en la experiencia numérica de la subsección 3.5.2. *Continúa en la página siguiente.*

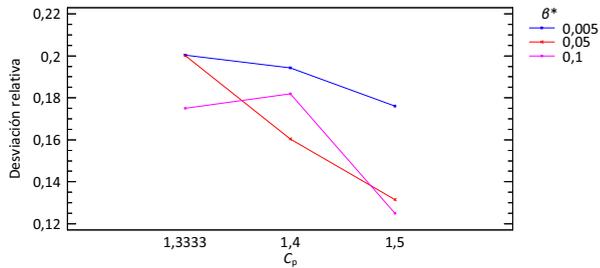
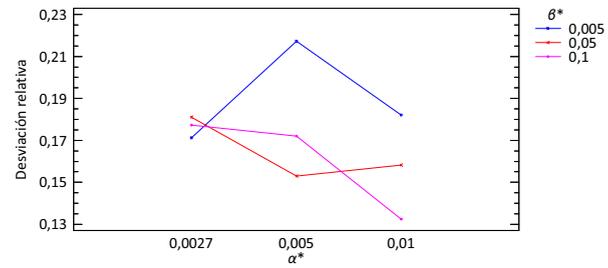
(e) Interacción $C_p \cdot \beta^*$.(f) Interacción $\alpha^* \cdot \beta^*$.

Figura 3.12: Gráficos de medias para las interacciones de segundo orden del ANOVA correspondiente a la comparación de los algoritmos considerados en la experiencia numérica de la subsección 3.5.2. *Continúa en la página siguiente.*

Grupo/algorithmo*	Enfoque [†]	Desviación (%) [‡]						Sol. (%) [§]	Mej. sol. (%) [¶]
		Media	Mín.	Q1	Mediana	Q3	Máx.		
Grupo A		0,67	0,00	0,00	0,00	0,00	10,63	100	78
GAMS/AlphaECP(sulum, pathnlp)	ECP	1,61	0,00	0,00	0,00	2,33	10,63	100	59
GAMS/Bonmin B-BB	B&B	0,22	0,00	0,00	0,00	0,00	5,33	100	89
GAMS/DICOPT(..., snopt)	OA	0,00	0,00	0,00	0,00	0,00	0,00	100	100
GAMS/KNITRO	B&B	0,77	0,00	0,00	0,00	0,00	6,84	100	78
GAMS/LINDOGlobal	B&B	0,91	0,00	0,00	0,00	1,42	6,20	100	67
GAMS/SBB(conopt)	B&B	0,52	0,00	0,00	0,00	0,00	5,33	100	78
Grupo B		15,31	0,00	7,23	16,07	22,00	50,50	100	16
GAMS/Bonmin B-OA	OA	13,39	0,00	0,00	7,23	22,50	50,50	100	30
GAMS/Bonmin B-QG	QG	16,22	9,75	13,40	16,63	18,46	24,20	100	0
GAMS/OQNLP(snopt)	OQNLP	16,33	0,00	5,74	19,25	26,06	34,75	100	19
Grupo C		43,29	0,00	18,08	47,36	62,21	130,39	71	19
Excel Solver	B&B	51,40	0,00	0,00	69,11	86,31	130,39	37	37
GAMS/Bonmin B-ECP	QG	40,29	0,00	0,94	31,83	73,44	104,75	56	26
GAMS/Bonmin B-Hyb	B&B + QG	37,83	0,00	31,79	41,35	50,70	69,11	96	11
GAMS/Bonmin B-iFP	FP + OA	43,65	6,00	37,79	47,33	51,18	69,11	96	0

—

* Véase más información sobre los algoritmos en la tabla 3.1.

† B&B = Bifurcación y acotación. ECP = Plano de corte extendido. FP = *Feasibility pump*. OA = Aproximación exterior. OQNLP = OptQuest/NLP. QG = *Branch-and-bound* no lineal de Quesada y Grossmann. Todos estos enfoques han sido presentados en la sección 1.4.2.

‡ Estadísticos descriptivos de la desviación (tras asignar valores a las ejecuciones sin solución) dentro de cada grupo o algoritmo. Media = Valor promedio. Mín. = Valor mínimo observado. Q1 = Primer cuartil. Mediana = Valor de la mediana. Q3 = Tercer cuartil. Máx. = Valor máximo observado.

§ Porcentaje de ejecuciones en que el algoritmo (o el grupo de algoritmos) devuelve una solución válida.

¶ Porcentaje de ejecuciones en que el algoritmo (o el grupo de algoritmos) devuelve la mejor solución conocida.

Tabla 3.8: Resumen del comportamiento de los tres grupos de algoritmos identificados en la segunda fase de la experiencia numérica de la subsección 3.5.2. Se destacan en negrita el mejor y el peor algoritmo.

(43,29 %). Estos y otros datos relevantes de los tres grupos se recogen en la tabla 3.8.

Tal y como se muestra en dicha tabla, el algoritmo con un mejor comportamiento es GAMS/DICOPT(...,snopt) —es decir, GAMS/DICOPT configurado con GAMS/BDMLP, GAMS/CBC, GAMS/CPLEX, GAMS/GUROBI, GAMS/XA o GAMS/XPRESS como *solver* auxiliar de MILP y GAMS/SNOPT como *solver* de NLP—, como ya se podía intuir de los resultados obtenidos en su calibración, y como también se ha visto en el análisis numérico realizado anteriormente. Dicho algoritmo ha devuelto la mejor solución conocida para toda la batería de 27 problemas. Incluso otras variantes de GAMS/DICOPT presentan desviaciones máximas de sólo $1,7 \cdot 10^{-3}$ % (es el caso de GAMS/DICOPT(xa,ipopt), por ejemplo).

Por contra, el algoritmo que ha ofrecido un peor comportamiento es la herramienta Solver de Excel, con una desviación media de 51,40 %. Sólo devolvió solución en el 37 % de las ejecuciones, aunque en tales casos siempre se trató de la mejor conocida. Otros dos algoritmos que destacan por no alcanzar en *ninguna* ocasión la mejor solución conocida, a pesar de devolver siempre o casi siempre alguna, son GAMS/Bonmin B-QG y GAMS/Bonmin B-iFP, lo cual ya se había mencionado en el análisis numérico inicial.

El ANOVA no detecta efectos significativos de los parámetros del problema sobre la calidad de la solución final (p -valores de 0,0689, 0,5102 y 0,3134 para C_p , α^* y β^* , respectivamente; figuras 3.11(b)–(d)). En el plano puramente muestral, se observa que, en general, valores menos restrictivos de los parámetros favorecen la consecución de mejores soluciones.

En cuanto a las interacciones de segundo orden, el ANOVA únicamente considera significativas las existentes entre los algoritmos y los parámetros α^* y β^* (p -valores de 0,0019 y menor que 0,0001, respectivamente; figuras 3.12(b) y 3.12(c)). En el caso concreto del parámetro β^* , éste actúa diluyendo parcialmente la separación de los algoritmos en tres grupos de eficacia mencionada anteriormente, si bien a nivel muestral se sigue observando cómo el grupo A de algoritmos proporciona los mejores

resultados, para cualquier valor de β^* que se considere.

Análisis del caso $C_p = 1,2$ A continuación se aborda el análisis de los 4 problemas de la batería con $C_p = 1,2$ para los que algún algoritmo proporcionó solución (ejemplos 6 a 9). Básicamente se reproducen los pasos seguidos arriba con los 27 problemas con $C_p \in \{1,3333; 1,4; 1,5\}$ (es decir, primero se calibran los algoritmos con más de una variante y después se comparan los algoritmos entre sí), si bien en este caso las unidades experimentales (es decir, los 4 problemas) serán consideradas como un único factor de bloqueo.

Un total de 63 (28,1 %) de las $8 \times 7 \times 4 = 224$ experiencias realizadas con GAMS/AlphaECP sobre esta subbatería no devolvieron solución. Se trata, en concreto, de todas las ejecuciones que usaban GAMS/GUROBI como *solver* de MILP auxiliar sobre el problema 6, todas las ejecuciones sobre el problema 7 excepto las que usaban GAMS/BDMLP como algoritmo de MILP, y todas las ejecuciones sobre el problema 8 configuradas con el método de MINLP GAMS/XA. De las ejecuciones que devolvieron solución, 55 (un 34,2 %) devolvieron la mejor solución conocida. En el resto (ejecuciones que devolvieron una solución peor que la mejor conocida), la desviación media fue de 24,74 %. Tras asignar valores a las ejecuciones que no devolvieron solución, la desviación promedio del total de ejecuciones de GAMS/AlphaECP sobre este conjunto de problemas fue 18,24 %.

En el caso de las experiencias con GAMS/DICOPT, un 30,4 % no devolvieron solución, mayoritariamente (94,1 %) asociadas al uso de GAMS/IPOPT o GAMS/MSNLP como método de NLP. Todas las ejecuciones con solución (156) devolvieron la mejor solución conocida excepto una. La desviación media tras asignar valores a las experiencias sin solución fue 14,47 %.

Las 7 ejecuciones de GAMS/OQNLP sobre el ejemplo 7 de la batería no fueron capaces de devolver solución. En el resto, únicamente las 7 ejecuciones sobre el problema 9 alcanzaron la solución óptima. En las experiencias que devolvieron una solución diferente de la óptima (un total de 14) la desviación media fue de 17,69 %. Tras completar las

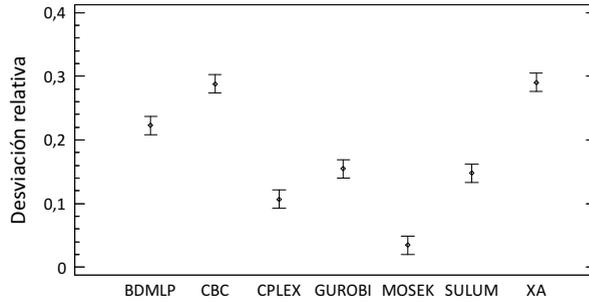
observaciones faltantes, la desviación promedio global producida por GAMS/OQNLP fue de 12,33 %.

Por último, sólo 3 (10,7 %) de las 28 ejecuciones de GAMS/SBB sobre los problemas 6 a 9 no devolvieron solución, todas ellas asociadas al algoritmo auxiliar de NLP GAMS/MINOS. Del resto, ninguna experiencia logró alcanzar la mejor solución conocida; la desviación promedio en este grupo fue de 2,52 %. La desviación promedio global después de asignar los valores faltantes fue de 8,42 %.

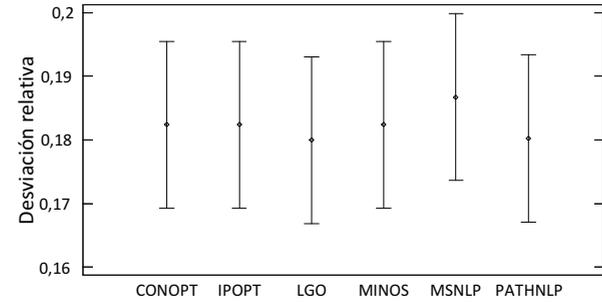
Las tablas A.18 a A.27 del anexo A recogen los resultados de los respectivos ANOVA de calibración de los algoritmos mencionados y los cálculos de todas las correspondientes medias. En el caso de GAMS/OQNLP y GAMS/SBB, el análisis no permite estimar el efecto de las interacciones ya que sólo se dispone de una observación para cada una de las combinaciones de orden dos.

El ANOVA para la calibración del algoritmo GAMS/AlphaECP sobre esta subbatería de ejemplos (tabla A.18 del anexo A) revela que la elección del *solver* auxiliar de MILP influye significativamente (p -valor $< 0,0001$) en la eficacia del algoritmo; no así el método de NLP (p -valor = 0,9928). El método de MILP que ofrece un mejor resultado es GAMS/MOSEK (figura 3.13(a)). Las unidades experimentales resultan tener también un efecto significativo en calidad de la solución, tanto como factor principal (p -valor $< 0,0001$; figura 3.13(c)) como en su interacción con el método auxiliar de MILP (p -valor $< 0,0001$; figura 3.14(b)), lo cual parece consecuencia directa de dónde se concentran las ejecuciones sin solución (véase más arriba).

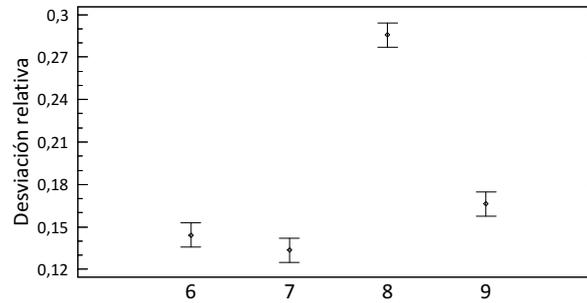
A la vista de estos datos y de la información recogida en la tabla A.20 del anexo A, la elección preferible para GAMS/AlphaECP sería utilizar GAMS/MOSEK como método MILP auxiliar (desviación media de 3,48 % en todos los casos, con igual desviación e igual porcentaje de ejecuciones con la mejor solución). Independientemente del método de NLP elegido, dicha configuración proporciona siempre en nuestra subbatería la mejor solución conocida, excepto para el problema 7, caso en que las únicas configuraciones que devuelven solución utilizan GAMS/BDMLP, como



(a) Efecto del método de MINLP.

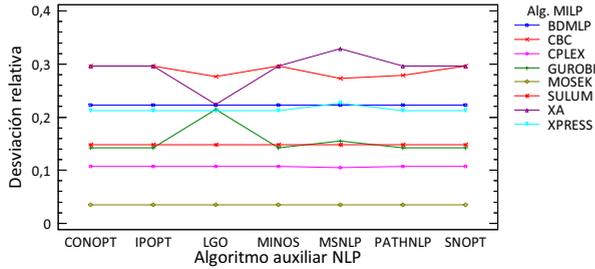


(b) Efecto del método de NLP.

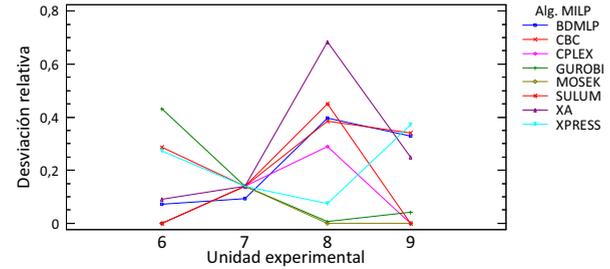


(c) Efecto de las unidades experimentales.

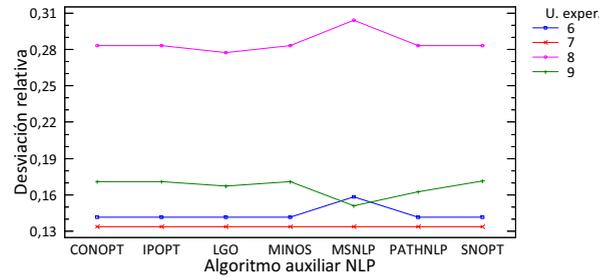
Figura 3.13: Gráficos de medias e intervalos HSD para los efectos principales del ANOVA correspondiente a la calibración del algoritmo GAMS/AlphaECP para el caso $C_p = 1,2$.



(a) Interacción MILP·NLP.



(b) Interacción MILP· unidades experimentales.



(c) Interacción NLP· unidades experimentales.

Figura 3.14: Gráficos de medias para las interacciones de segundo orden del ANOVA correspondiente a la calibración del algoritmo GAMS/AlphaECP para el caso $C_p = 1,2$.

ya se ha mencionado.

La calibración de GAMS/DICOPT (tabla ANOVA [A.21](#) del anexo [A](#)) revela que tanto el método de MILP como el de NLP que se elijan influyen en la eficacia del algoritmo (p -valores 0,0088 e inferior a 0,0001, respectivamente; figuras [3.15\(a\)](#) y [3.15\(b\)](#)), sin interacciones significativas entre ellos (p -valor = 0,6212; figura [3.16\(a\)](#)). Al igual que sucedía en el caso general, estudiado anteriormente, esto parece tener relación directa con el hecho de que las experiencias en que el algoritmo no devuelve solución están asociadas al uso de ciertos algoritmos auxiliares en concreto (véase más arriba), junto a que en prácticamente el resto de ejecuciones se alcanza la mejor solución conocida (desviación nula).

A partir de la información contenida en la tabla de medias [A.23](#) del anexo [A](#), puede concluirse que la mejor configuración de GAMS/DICOPT cuando opera sobre problemas de OPC similares a los de la subbatería es utilizar GAMS/BDMLP, GAMS/CBC, GAMS/CPLEX, GAMS/GUROBI, GAMS/XA o GAMS/XPRESS como método de MILP auxiliar (cualquiera excepto GASM/MOSEK y GAMS/SULUM) junto con GAMS/CONOPT, GAMS/LGO, GAMS/MINOS, GAMS/PATHNLP o GAMS/SNOPT como método de NLP (todos excepto GAMS/IPOPT y GAMS/MSNLP). Cualquiera de esas $6 \times 5 = 30$ variantes proporciona siempre la mejor solución conocida en todos los ejemplos objeto de estudio en este apartado (es decir, ofrece una desviación media de 0%).

El ANOVA realizado para calibrar GAMS/OQNLP sobre esta batería de 4 problemas (tablas [A.24](#) y [A.25](#) del anexo [A](#); figuras [3.17\(a\)](#) y [3.17\(b\)](#)) permite concluir que todos los métodos auxiliares de NLP proporcionan resultados similares (p -valor = 0,4412). En el plano muestral, GAMS/IPOPT y GAMS/MSNLP proporcionan peores resultados, únicamente debido a que funcionan peor con el problema 8 de la batería. Así, la recomendación sería configurar el algoritmo con uno de los siguientes métodos auxiliares de NLP: GAMS/SNOPT (desviación media de 11,4275 % en la muestra) o GAMS/CONOPT, GAMS/LGO, GAMS/MINOS y GAMS/PATHNLP (desviación media de 11,4276 %).

El análisis para la calibración del algoritmo GAMS/SBB (tablas [A.26](#) y

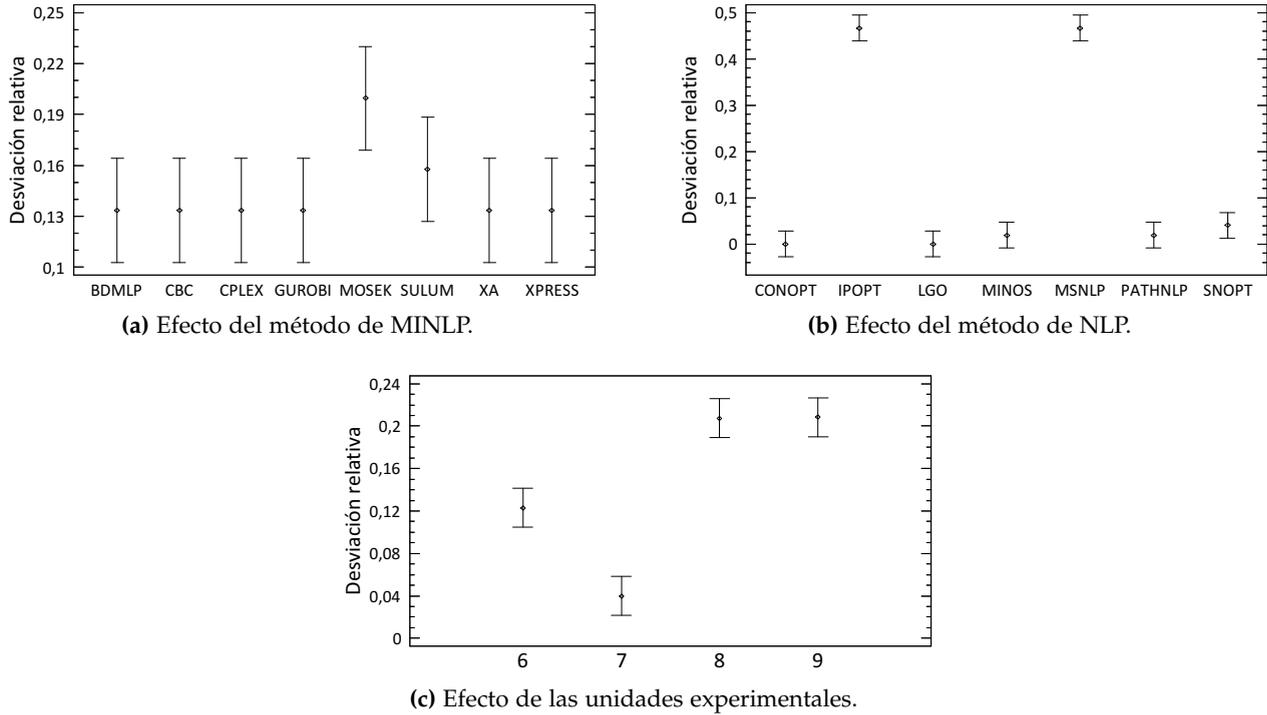
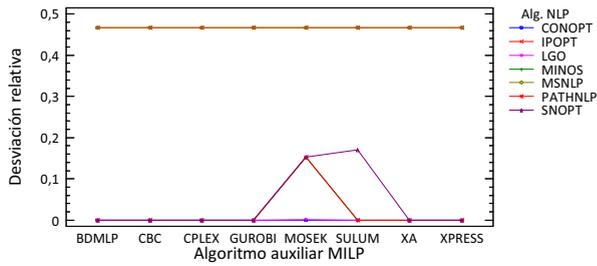
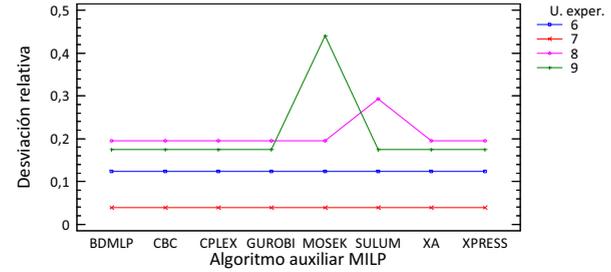


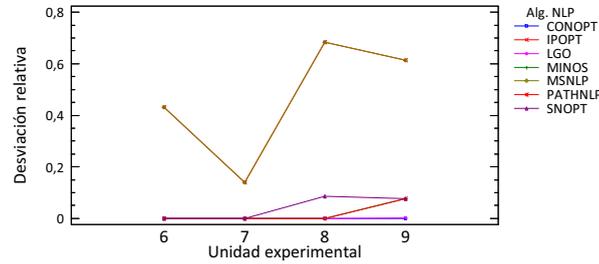
Figura 3.15: Gráficos de medias e intervalos HSD para los efectos principales del ANOVA correspondiente a la calibración del algoritmo GAMS/DICOPT para el caso $C_p = 1,2$.



(a) Interacción MILP·NLP.

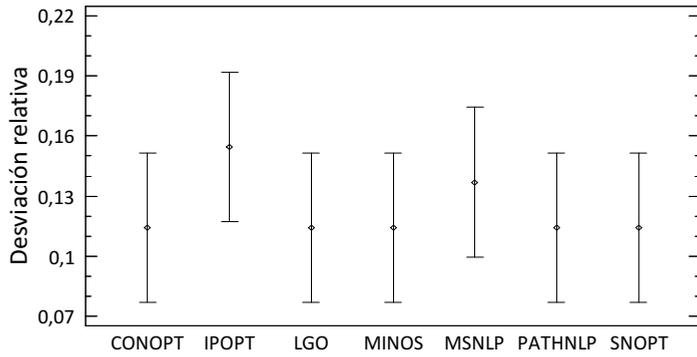


(b) Interacción MILP· unidades experimentales.

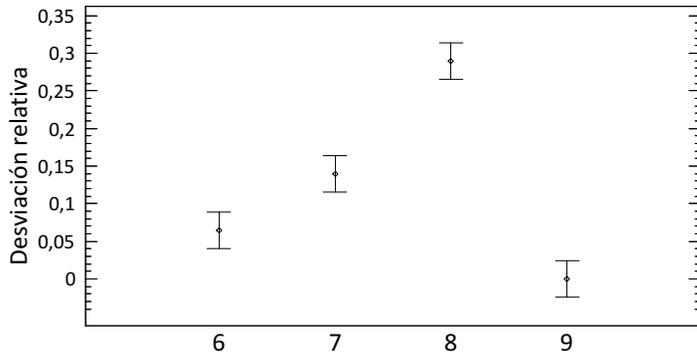


(c) Interacción NLP· unidades experimentales.

Figura 3.16: Gráficos de medias para las interacciones de segundo orden del ANOVA correspondiente a la calibración del algoritmo GAMS/DICOPT para el caso $C_p = 1,2$.



(a) Efecto del método de NLP.



(b) Efecto de las unidades experimentales.

Figura 3.17: Gráficos de medias e intervalos HSD para los efectos principales del ANOVA correspondiente a la calibración del algoritmo GAMS/OQNLP para el caso $C_p = 1,2$.

A.27 del anexo A; figuras 3.18(a) y 3.18(b)) revela lo que ya se observaba a nivel muestral. Cuando se abordan problemas como los de la subbatería, puede configurarse el algoritmo GAMS/SBB con cualquier solver de NLP auxiliar excepto GAMS/MINOS. Éste (como se vio arriba) no devuelve solución en 3 de sus 4 ejecuciones, mientras que el resto de algoritmos alcanzan todos la misma solución en cada uno de los 4 problemas objeto de interés, excepto la ejecución con GAMS/PATHNLP sobre el problema 9, que proporciona una desviación unas 21 veces mayor que el resto

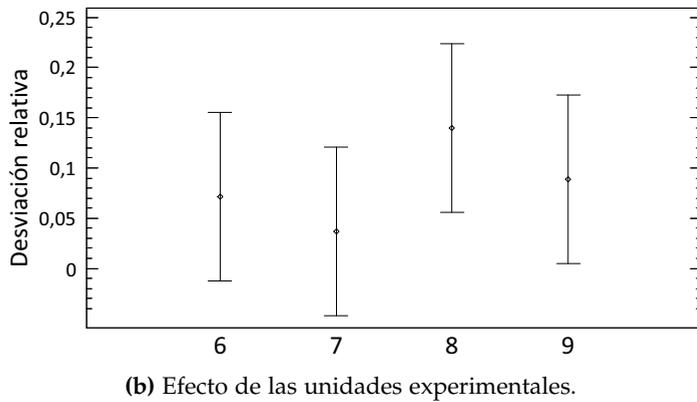
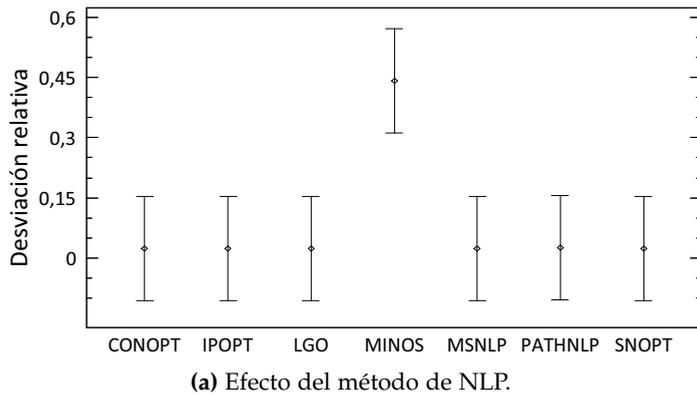


Figura 3.18: Gráficos de medias e intervalos HSD para los efectos principales del ANOVA correspondiente a la calibración del algoritmo GAMS/SBB para el caso $C_p = 1,2$.

para ese problema, aunque aceptable en cualquier caso: 0,808 % frente a 0,038 %. La desviación media conseguida usando GAMS/CONOPT, GAMS/IPOPT, GAMS/LGO, GAMS/MSNLP o GAMS/SNOPT como métodos auxiliares es de 2,440 % en la muestra estudiada.

La tabla 3.9 resume las conclusiones de la calibración que se acaba de realizar de los cuatro algoritmos con más de una variante para el caso particular de $C_p = 1,2$.

Las conclusiones que se obtienen no son muy distintas a las del caso

Algoritmo* (versión [†])
GAMS/AlphaECP(mosek,*)
GAMS/DICOPT(bdmlp cbc cplex gurobi xa xpress, conopt lgo minos pathnlp snopt)
GAMS/OQNLP(conopt lgo minos pathnlp snopt)
GAMS/SBB(conopt ipopt lgo msnlp snopt)

—

* Véase más información sobre los algoritmos en la tabla 3.1.
[†] Un asterisco sirve para indicar que cualquier opción es válida, y una barra vertical para separar opciones igualmente válidas.

Tabla 3.9: Resumen de la calibración realizada para los algoritmos con más de una variante en la experiencia numérica de la subsección 3.5.2 para el caso $C_p = 1,2$. Continuando y extendiendo la notación introducida en dicha subsección, entre paréntesis se indican los algoritmos auxiliares de elección.

general. El algoritmo GAMS/AlphaECP es en el que más difieren los resultados: mientras que en el caso general (para $C_p \in \{1,3333; 1,4; 1,5\}$) el método auxiliar de MILP de elección debía ser GAMS/SULUM, seguido de GAMS/XA y GAMS/GUROBI, en el caso de $C_p = 1,2$ es mejor configurar GAMS/AlphaECP con GAMS/MOSEK como *solver* de MILP.

Las 6 variantes de GAMS/DICOPT con mejor eficacia en el caso general (utilizando cualquier *solver* auxiliar de MILP excepto GAMS/MOSEK y GAMS/SULUM, junto con GAMS/SNOPT como *solver* de NLP) están incluidas entre las 36 que resultan *ganadoras* en el caso $C_p = 1,2$. La diferencia más notable se produce el comportamiento de GAMS/IPOPT como algoritmo auxiliar de NLP: en el caso general está asociado a altas desviaciones medias, mientras que en el caso $C_p = 1,2$ se encuentra entre los algoritmos de elección.

GAMS/OQNLP genera sus mejores resultados cuando es configurado con GAMS/SNOPT como técnica de NLP auxiliar, tanto en el caso general como en el particular. Otros algoritmos de NLP que son perfectamente válidos en ambos casos son GAMS/CONOPT, GAMS/LGO y GAMS/PATHNLP. GAMS/MINOS no es recomendable en el caso general, mientras que GAMS/IPOPT no lo es en el particular. GAMS/MSNLP se considera a evitar en los dos casos.

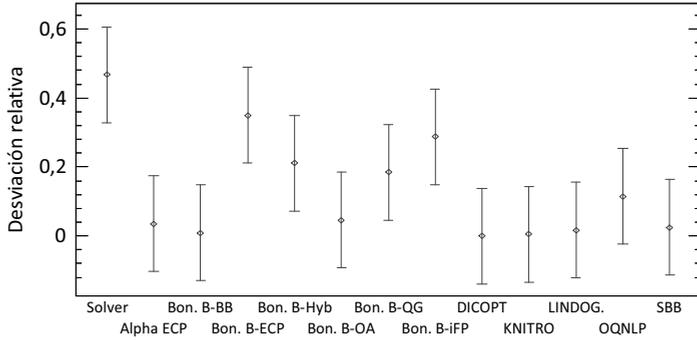
Por último, la configuración óptima de GAMS/SBB en la muestra

general resultó ser con GAMS/CONOPT como *solver* auxiliar de NLP, lo cual encaja con la recomendación que acabamos de ver para el caso particular, en el que todos los métodos excepto GAMS/MINOS son igualmente de elección. Este último sí que es válido para el caso general; no así GAMS/MSNLP, GAMS/PATHNLP y GAMS/SNOPT, como se ha visto.

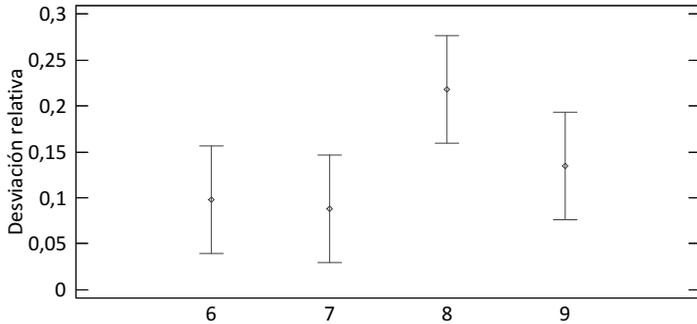
Tras esto, al igual que en el caso general, incorporamos el resto de algoritmos tomados en consideración y realizamos un ANOVA con los 13 algoritmos base como factor objeto de interés y la subbatería de problemas como factor de bloqueo (tablas A.28 y A.29 del anexo A; figuras 3.19(a) y 3.19(b)) para determinar si hay diferencias entre la calidad de las soluciones que proporciona cada método al problema del OPC. Como sucedió con la calibración de GAMS/OQNLP y GAMS/SBB para el caso $C_p = 1,2$, tampoco fue posible en este caso estudiar las interacciones, debido a que sólo se cuenta con una observación para cada combinación de algoritmo y problema.

De acuerdo con la figura 3.19(a) y la tabla A.29, el mejor y el peor algoritmo siguen siendo, como en el caso general antes estudiado, GAMS/DICOPT (desviación media de 0 %) y la herramienta Solver de Excel (desviación media de 46,68 %), respectivamente.

Los tres grupos A, B y C de algoritmos que se distinguían claramente en el caso general (de mejor a peor eficacia) se mantienen en este caso, aunque de manera más difusa. Se sigue cumpliendo que GAMS/AlphaECP, GAMS/Bonmin B-BB, GAMS/DICOPT, GAMS/KNITRO, GAMS/LINDOGlobal y GAMS/SBB ofrecen mejores resultados que el resto, y que GAMS/Bonmin B-ECP, GAMS/Bonmin B-Hyb, GAMS/Bonmin B-iFP y Solver son los que se comportan peor, si bien el orden relativo entre ellos varía un poco. Además, el algoritmo GAMS/Bonmin B-OA, que en el caso general estaba ubicado en el grupo B (de eficacia media) junto con GAMS/Bonmin B-QG y GAMS/OQNLP, en este caso queda más cerca de los algoritmos con menor desviación media. De igual forma, Bonmin B-QG, también en el grupo B, ofrece una desviación media sobre la subbatería más próxima a la de los algoritmos



(a) Efecto del algoritmo.



(b) Efecto de las unidades experimentales.

Figura 3.19: Gráficos de medias e intervalos HSD para los efectos principales del ANOVA correspondiente a la comparación de los algoritmos considerados en la experiencia numérica de la subsección 3.5.2, para el caso $C_p = 1,2$.

del grupo C.

Conclusiones de la experiencia numérica

Acabamos de mostrar a través de un ejemplo concreto la utilidad de nuestra propuesta, consistente en plantear como problema de MINLP la configuración óptima de la técnica del precontrol, visto como herramienta sencilla para detectar situaciones de falta de control, adaptado a las exigencias del usuario final en términos de máxima tasa de falsa alarma

deseada y mínima potencia para detectar una desviación inaceptable del proceso.

Como resumen de todo lo analizado en la experiencia numérica podemos destacar los siguientes hechos:

1. El algoritmo que mejor comportamiento ha ofrecido en términos de eficacia al resolver problemas de OPC ha sido GAMS/DICOPT, especialmente cuando es configurado con GAMS/SNOPT como método de NLP y GAMS/BDMLP, GAMS/CBC, GAMS/CPLEX, GAMS/GUROBI, GAMS/XA o GAMS/XPRESS como método de MILP, combinación que, para todos los problemas de la batería excepto los supuestamente no factibles, proporciona siempre la mejor solución conocida.
2. Otros algoritmos con una eficacia similar a GAMS/DICOPT, según ha revelado el análisis estadístico, son (sin que el orden sea relevante) GAMS/AlphaECP (particularmente con GAMS/SULUM, GAMS/XA o GAMS/GUROBI como *solver* de MILP), GAMS/BONMIN B-BB, GAMS/KNITRO, GAMS/LINDOGlobal y GAMS/SBB (usando GAMS/CONOPT como método auxiliar de NLP); véanse más detalles en la tabla 3.8, referida a los ejemplos 10 a 36 de la batería.
3. El algoritmo con un peor comportamiento en cuanto a eficacia ha sido la herramienta Solver de Excel (que, como se ha detallado en la tabla 3.1, implementa una estrategia de tipo B&B), sobre todo por la elevada cantidad de casos en que se detuvo a causa de un error de ejecución (para un 68 % de los ejemplos de la batería, excluyendo los problemas supuestamente no factibles), si bien cuando convergió a una solución siempre lo hizo a la mejor conocida, en la batería estudiada.
4. Según el ANOVA realizado, los algoritmos GAMS/Bonmin B-iFP, GAMS/Bonmin B-Hyb y GAMS/Bonmin B-ECP presentan una eficacia insuficiente, similar a la de Solver; en el caso de los

dos primeros, por alcanzar en pocas o ninguna ocasión la mejor solución conocida de los problemas de la batería; y en el caso de GAMS/Bonmin B-ECP, por presentar además, como Solver, un elevado porcentaje de ejecuciones con error (véase la tabla 3.8 para más detalles referidos a los ejemplos 10 a 36 de la batería).

5. La elección de una determinada configuración en aquellos algoritmos con más de una variante puede conducir a resultados muy distintos en términos de eficacia. Por ejemplo, GAMS/DICOPT ofrece en general soluciones de alta calidad, como se ha visto, excepto cuando es configurado con GAMS/MSNLP como método de NLP, caso en el cual produjo un 97 % de ejecuciones con error sobre la batería de los ejemplos 6 a 36 (véase la tabla A.2). En el caso de GAMS/AlphaECP, la diferencia entre la combinación con menor y mayor desviación media, ejecutadas sobre los ejemplos 10 a 36 de la batería, es de alrededor de 15,7 veces (desviación de 1,61 % para la combinación SULUM,PATHNLP frente a 25,26 % con CBC,CONOPT; véase la tabla A.6).

6. Junto con lo anterior, y tal y como se desprende de la documentación de los algoritmos, de las pruebas preliminares realizadas y ya mencionadas en su momento, del hecho de haber encontrado en la experiencia numérica muchas ejecuciones que no lograron devolver una solución factible cuando sí la había y del hecho de que ninguna de las ejecuciones con cualquiera de los algoritmos llegó a garantizar ni la convergencia a un óptimo global ni, en su caso, la infactibilidad, puede concluirse que el resultado final de los algoritmos de MINLP probados es altamente dependiente de la manera concreta de modelizar el problema a resolver y del punto de arranque que se les proporcione, así como de la calibración que se realice de los diferentes parámetros y opciones de configuración que cada uno de ellos ofrece. Esto, en nuestra opinión, dificulta en la práctica su empleo para la resolución sistemática de problemas de OPC por parte de usuarios no avanzados.

7. El valor asignado a los parámetros del problema (C_p , α^* y β^*) influye en la calidad media de las soluciones que los algoritmos de MINLP son capaces de encontrar; es decir, hay problemas de OPC que son significativamente más sencillos o difíciles de resolver que otros. Incluso, para algunos valores de dichos parámetros, algunos algoritmos presentan una eficacia diferente a la esperada. Sin embargo, esto no invalida ni modifica sustancialmente ninguna de las conclusiones precedentes; en todo caso, refuerza la conclusión anterior.

Lo dicho, como es obvio, se refiere a los algoritmos de MINLP que se han incluido en la experiencia (véase la tabla 3.1), pero puede generalizarse, en todo o en gran parte, a cualquier algoritmo de MINLP de propósito general. Dichos algoritmos suelen precisar de una calibración *inteligente* de sus diferentes opciones, muchas veces realizada por el procedimiento de *prueba y error*. Junto a esto, las condiciones específicas del problema a resolver y la manera concreta de modelizarlo influyen en la probabilidad de éxito del algoritmo (no sólo ya en MINLP, sino incluso en programación lineal).

Todo esto, referido a un problema como el del OPC (3.34) que, si bien es no lineal y entero, únicamente está compuesto por dos restricciones y tres variables, hace pensar que *debe* haber alguna forma *más fiable* de resolverlo. En las conclusiones finales del capítulo se desarrolla esta idea.

3.5.3. Análisis del peor caso

Seguidamente aprovechamos la experiencia numérica cuyos resultados terminamos de analizar para realizar un estudio descriptivo de lo que podría llamarse el *peor caso* en relación a los planes de precontrol óptimos.

Sabemos que los planes de precontrol obtenidos en la experiencia numérica son óptimos (o al menos, tan buenos como han sido capaces de asegurar los algoritmos utilizados), en el sentido de que minimizan (o hacen aceptablemente pequeño) el número esperado de piezas a muestrear *cuando el proceso se encuentra centrado* (véase la definición de la función

α^*	β^*	$(k, t, \lambda)_{\text{opt}}$	$E[n(0)]_{\text{opt}}$	$\max_{\delta} E[n(\delta)]$	Var. sobre $E[n(0)]$	$\delta_{\text{máx}}$	$E[n(\delta^*)]$	Var. sobre $E[n(0)]$
0,0027	0,005	(7, 4, 5,096)	11,81	33,67	185,1 %	1,235	11,287	-4,4 %
0,0027	0,05	(4, 5, 5,898)	6,612	19,261	191,3 %	1,434	11,654	76,3 %
0,0027	0,1	(3, 5, 6,46)	4,961	12,601	154 %	1,497	9,516	91,8 %
0,01	0,005	(6, 4, 5,596)	11,091	26,806	141,7 %	1,173	9,374	-15,5 %
0,01	0,05	(3, 5, 7,368)	5,909	12,902	118,3 %	1,346	8,49	43,7 %
0,01	0,1	(3, 4, 6,1)	4,613	9,603	108,2 %	1,455	7,379	60 %

Tabla 3.10: Estudio del peor caso para una selección de los planes de precontrol generados con $C_p = 1,3333$ y $\delta^* = 1,946$ (δ^* tal que $p^* = 2\%$).

objetivo en la subsección 3.3.3). Ahora bien, si el proceso presentase cierta desviación $\delta > 0$, podría ocurrir que el plan de precontrol obtenido obligase a realizar un muestreo de más piezas de las esperadas, para poder cualificar o no el proceso. Es decir, tal y como se apuntaba al finalizar la subsección 3.3.3, una vez fijado un plan de precontrol por este procedimiento, es pertinente estudiar el comportamiento de la expresión $E[n(\delta)]$ en función de la desviación estandarizada del proceso δ , con el fin de comprobar si el tamaño muestral puede alcanzar cotas inaceptables para algún valor de dicha desviación. Se trata de un estudio del *peor caso*.

En la tabla 3.10 se muestran, para una muestra de los planes de precontrol obtenidos en la experiencia numérica, el valor esperado del tamaño muestral cuando la desviación estandarizada δ es nula, el peor valor (el más alto) que puede tomar dicha esperanza matemática (véanse las expresiones (3.19) y (3.21)), y para qué valor de δ se alcanza. También se añade el tamaño muestral que será necesario, en término medio, para finalizar el procedimiento cuando δ toma el valor δ^* , la desviación inaceptable que se ha considerado al construir el plan de precontrol óptimo. Los cálculos se realizaron con Mathematica 9.0.1 para Windows (Wolfram, 2014).

En la figura 3.20 mostramos gráficamente el comportamiento de $E[n(\delta)]$ en función de δ , en planes de precontrol óptimos obtenidos para procesos con diferentes capacidades potenciales. El gráfico permite observar el aumento que se produce en el número esperado de piezas

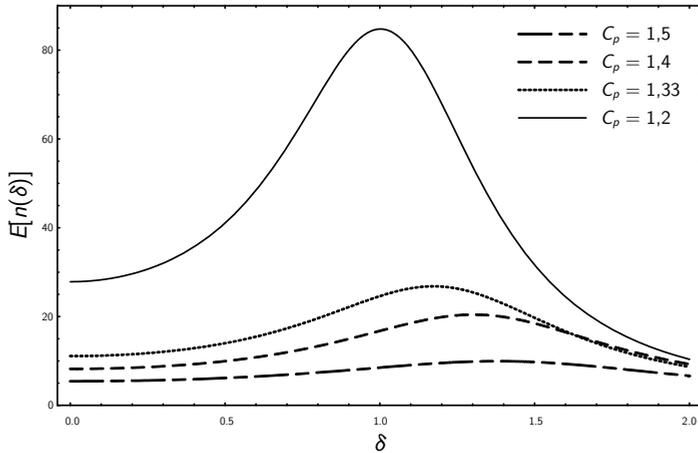


Figura 3.20: Curva de $E[n(\delta)]$ en función de δ , fijado el plan de precontrol óptimo, para $\alpha^* = 1\%$, $\beta^* = 0,5\%$ y δ^* tal que $p^* = 2\%$.

a muestrear cuando el proceso está *ligeramente* descentrado, y cómo, en procesos menos capaces (es decir, con mayor variabilidad), dicho aumento es más pronunciado.

Se observa cómo, en los casos estudiados, el tamaño muestral esperado requerido para la aplicación de la técnica puede llegar a ser, en ocasiones, de hasta casi 3 veces el tamaño medio necesario cuando el proceso se encuentra centrado (véase la tabla 3.10). Este hecho es menos grave de lo que puede parecer. Es esperable —y, hasta cierto punto, inevitable— que, cuando la desviación del proceso δ se aleja de una desviación nula, pero sin llegar a producir un alto porcentaje de piezas fuera de especificaciones, tenga que muestrearse un elevado número de piezas para que el precontrol finalice, ya que, en tal caso, se alternarán con frecuencia las extracciones de piezas verdes y amarillas.

Como se ha apuntado previamente en la subsección 3.3.3, intentar que dicho máximo sea lo menor posible nos llevaría a estudiar un problema de programación semiinfinita. Alternativamente, sería posible plantearse un enfoque *intermedio*, como, por ejemplo, añadir una restricción sobre el número esperado de piezas a muestrear cuando la media del proceso se

sitúe alrededor de la frontera entre las zonas verde y amarilla, de manera que dicho valor no sea mucho mayor que $E[n(0)]$. De todas maneras, es probable que lo único que consiguiéramos fuese (lógicamente) aumentar el valor de $E[n(0)]$, sin que disminuyera mucho el máximo número esperado de piezas a muestrear. Esta y otras posibles alternativas se discuten en el capítulo de conclusiones y líneas futuras.

3.6. Conclusiones

En este capítulo hemos desarrollado una metodología para la obtención de planes de precontrol óptimos, mediante el uso de técnicas de MINLP. Nuestra propuesta supone un paso adelante en la dirección señalada por [San Matías et al. \(2004\)](#) con la flexibilización de los parámetros que rigen el comportamiento de la técnica, ya que conseguimos optimizar y automatizar la elección de dichos parámetros (véase la figura 3.21).

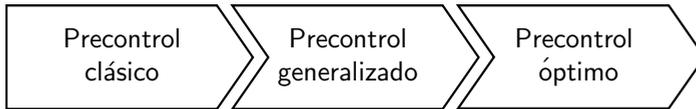


Figura 3.21: Relación del precontrol óptimo con las propuestas anteriores.

Dicha automatización supone la principal ventaja de nuestra propuesta, a la cual hay que añadir que permite mejorar el comportamiento y la potencia de la técnica clásica, sin perder las características que la hacían atractiva (esto es, la sencillez de uso, la rapidez de aplicación, y la relación directa con las especificaciones del producto).

Asimismo, el enfoque de optimización que hemos introducido, y que constituye en sí mismo la principal novedad en nuestra aportación a la técnica del precontrol, deja la puerta abierta a otras alternativas y mejoras, en la misma dirección.

Mediante una experiencia numérica hemos ilustrado la aplicación de nuestro método, y hemos estudiado para un caso particular el aumento del tamaño muestral ante desviaciones intermedias del proceso.

A la vista de los datos, es correcto afirmar que la generación de planes de precontrol óptimos es sencilla y asegura un comportamiento de la técnica dentro de los requerimientos deseados por el usuario; y con todo ello, puede concluirse que se ha alcanzado uno de los objetivos de este trabajo, como es plantear formalmente la determinación de los parámetros del precontrol como problema de optimización.

Como se ha puesto de manifiesto en la experiencia numérica llevada a cabo, la elección de una técnica de resolución para el problema (3.34), lejos de ser trivial, se convierte en un asunto fundamental dentro de la puesta en práctica de nuestra propuesta. A pesar de la aparente sencillez del modelo, se ha visto cómo diferentes técnicas pueden proporcionar resultados distintos e incluso no converger cuando otras sí dan una solución (véanse conclusiones más detalladas de la experiencia numérica en la subsección 3.5.2).

La alternativa a utilizar *software* de propósito general es, como se expuso ya en la subsección 3.5.1, el diseño de algoritmos específicos para el problema a resolver. Y en concreto, dadas las dimensiones del problema OPC (3.34), la primera opción *debe* ser intentar el desarrollo de un algoritmo específico con un enfoque *exacto*. A ello dedicaremos el capítulo 5. Como paso precedente, y con el fin de ampliar el conocimiento del mismo, en el capítulo 4 estudiaremos en profundidad las características y el comportamiento de las funciones del problema OPC (3.34).

Estudio de las propiedades del modelo de optimización del precontrol

En este capítulo abordamos el estudio analítico de las propiedades y el comportamiento de las expresiones involucradas en el que hemos llamado *problema del precontrol óptimo* (3.34) y que ha sido caracterizado como problema de MINLP en el capítulo anterior.

El propósito principal es profundizar en el conocimiento de dicho problema y utilizar los resultados que se obtengan en el diseño de algoritmos de resolución específicos. Es, por tanto, en el presente capítulo donde se concentra principalmente nuestra contribución teórica en relación al precontrol¹.

¹No obstante, tal y como ya se ha destacado en las conclusiones del capítulo 2, dedicado a presentar en detalle la técnica del precontrol y a revisar las variantes y mejoras existentes en la literatura, la sección 2.5 incluye, como contribuciones originales propias, la demostración de que la expresión para la probabilidad de cualificar desarrollada por [San Matías et al. \(2004\)](#) está bien definida (proposición 2.1), así como la formalización del precontrol como contraste de hipótesis sobre la capacidad del proceso.

Pensamos que el lugar en que se ubican las mencionadas contribuciones dentro de la presente memoria es el más adecuado de cara a mantener la lógica y la estructura de la misma. En concreto, el contenido de la sección 2.5 permite definir con propiedad el problema del precontrol óptimo (3.34) en el capítulo 3.

4.1. Notación, objetivos y estructura

4.1.1. Notación

En la tabla 4.1 (página 204 y siguientes) se enumeran y describen las variables, funciones y demás parámetros y expresiones que se utilizarán en las secciones siguientes. Algunos de ellos han aparecido ya previamente, y otros surgen en el transcurso de este capítulo; se recopilan todos aquí con el fin de facilitar el seguimiento y la comprensión de las proposiciones que a continuación se van a demostrar.

En algunos casos adaptamos, modificamos o ampliamos la notación empleada anteriormente para conseguir una mayor coherencia y claridad dentro del presente capítulo. Concretamente:

- Denotaremos por $S(C_p, \delta)$ o simplemente S a la *probabilidad de que una pieza esté dentro de especificaciones*; es decir,

$$S = \Phi(\delta + 3C_p) - \Phi(\delta - 3C_p) \quad (4.1)$$

para el caso normal que estamos estudiando (véanse las definiciones de Φ (2.23), C_p (1.1) y δ (2.11)). Nótese que, por definición,

$$S = p_g + p_y \quad (4.2)$$

$$= 1 - p_r. \quad (4.3)$$

Nótese asimismo que

$$0 < S < 1, \quad (4.4)$$

ya que $0 < p_r < 1$ según (3.5).

- Denotaremos la probabilidad de que el precontrol cualifique un determinado proceso como p_Q (véase (2.12)).
- La probabilidad de no detectar una determinada desviación del proceso δ será denotada por β en lugar de β_δ , incluyendo δ dentro de la lista de argumentos de la función.

- Nos referiremos a la función objetivo del problema del precontrol óptimo como f , en lugar de $E[n(0)]$ (véase (3.25)), por ser esta notación más sencilla y bastante habitual en trabajos del área de optimización.
- En aquellas funciones cuyo valor depende de varios parámetros, distinguiremos entre los argumentos que *varían* y los que se consideran dados o constantes, y lo expresaremos mediante la notación «función(variables | constantes)». Dicha notación se adaptará al caso concreto que se esté estudiando, y en ocasiones se omitirán todos o algunos de los argumentos considerados *constantes* para ganar en claridad y simplicidad.

Por ejemplo, si consideramos la tasa de falsa alarma α como función de las variables del precontrol k , t y λ , la cual depende también del parámetro del proceso C_p , nos referiremos a ella como $\alpha(k, t, \lambda | C_p)$; mientras que si estamos abordando el caso en que consideramos fijados los parámetros k y t , tratando α como función de λ , lo denotaremos como $\alpha(\lambda | k, t, C_p)$, o simplemente $\alpha(\lambda)$. Como último ejemplo, si consideramos la tasa de falsa alarma como función de k cuando t está fijado a un valor arbitrario y λ toma el valor 2, nos referiremos a ella como $\alpha(k | t, \lambda = 2, C_p)$ o simplemente $\alpha(k | \lambda = 2)$.

Como ya se ha dicho, el problema de optimización en torno al que giran todas las demostraciones de este capítulo es el del *precontrol óptimo* o problema OPC, desarrollado en el capítulo 3 y recogido en la expresión (3.34). De acuerdo con los convenios de notación que acabamos de introducir, dicho problema quedaría expresado de la siguiente forma:

$$\left. \begin{array}{ll} \text{Min} & f(k, t, \lambda | C_p) \\ \text{s.a:} & \alpha(k, t, \lambda | C_p) \leq \alpha^* \\ & \beta(k, t, \lambda | C_p, \delta^*) \leq \beta^* \\ & k, t \geq 1, \lambda \geq 2 \\ & k, t \in \mathbb{Z}, \lambda \in \mathbb{R} \end{array} \right\}. \quad (4.5)$$

Símbolo	Significado	Definición
$\Phi(z)$	Función de distribución acumulada de una variable aleatoria normal tipificada unidimensional.	(2.23), p. 88
$\varphi(z)$	Función de densidad de una variable aleatoria normal tipificada unidimensional.	(4.13), p. 213
C_p	Capacidad potencial del proceso.	(1.1), p. 20
δ	Desviación estandarizada del proceso.	(2.11), p. 84
δ^*	Desviación inaceptable del proceso.	Sec. 3.3.4, p. 120
k	Primer parámetro del precontrol. Número necesario de verdes consecutivas para cualificar el proceso.	Sec. 2.4, p. 80
t	Segundo parámetro del precontrol. Número necesario de amarillas consecutivas para detener y revisar el proceso.	Sec. 2.4, p. 80
λ	Tercer parámetro del precontrol. Razón entre la anchura del rango de tolerancias y la <i>semianchura</i> de la zona verde.	(2.24), p. 88
$S(C_p, \delta)$	Probabilidad de obtener una observación dentro de especificaciones.	(4.1), p. 202
$p_g(\lambda C_p, \delta)$	Probabilidad de obtener una observación verde.	(2.21), p. 88
$p_y(\lambda C_p, \delta)$	Probabilidad de obtener una observación amarilla.	(2.22), p. 88
$p_r(C_p, \delta)$	Probabilidad de obtener una observación roja.	(3.4), p. 115
$p_Q(k, t, \lambda C_p, \delta)$	Probabilidad de que el procedimiento del precontrol termine cualificando el proceso.	(2.12), p. 84
$f(k, t, \lambda C_p)$	Función objetivo del problema OPC. Número esperado de piezas a muestrear para terminar el procedimiento del precontrol cuando la desviación δ es 0.	(3.25), p. 119

Tabla 4.1: Relación y significado de las expresiones matemáticas particulares empleadas en el presente capítulo. *Continúa en la página siguiente.*

Símbolo	Significado	Definición
$\alpha(k, t, \lambda \mid C_p)$	Probabilidad de falsa alarma. Lado izquierdo de la primera restricción del problema OPC.	(3.28), p. 120
$\beta(k, t, \lambda \mid C_p, \delta)$	Probabilidad de no detectar una desviación del proceso δ . Lado izquierdo de la segunda restricción del problema OPC.	(3.32), p. 121
α^*	Cota superior para la probabilidad de falsa alarma. Lado derecho de la primera restricción del problema OPC.	Sec. 3.3.1, p. 112
β^*	Cota superior para la probabilidad de no detectar una cierta desviación del proceso. Lado derecho de la segunda restricción del problema OPC.	Sec. 3.3.1, p. 112
$K_{\text{máx}}$	Cota superior preestablecida para el valor de k .	Sec. 3.5.2, p. 139
$T_{\text{máx}}$	Cota superior preestablecida para el valor de t .	Sec. 3.5.2, p. 139
$L_{\text{máx}}$	Cota superior preestablecida para el valor de λ .	Sec. 3.5.2, p. 139
OPC	Problema del precontrol óptimo.	(3.34), p. 122; (4.5), p. 203
OPC-b	Problema del precontrol óptimo acotado.	(3.36), p. 139; (4.6), p. 207
OPC-sb	Problema del precontrol óptimo semiacotado.	(4.7), p. 208
OPC-NLP	Problema auxiliar de optimización de λ .	(4.117), p. 267
RF_{OPCsb}	Región factible del problema OPC-sb.	(4.133), p. 279
$\lambda_\alpha(k, t)$	Si existe, valor de λ para el cual se cumple con igualdad la primera restricción del problema OPC, fijadas k y t .	Prop. 4.8, p. 268
$\lambda_\beta(k, t)$	Mínimo valor de λ para el cual se cumple la segunda restricción del problema OPC, fijadas k y t .	Prop. 4.9, p. 271

Tabla 4.1: Relación y significado de las expresiones matemáticas particulares empleadas en el presente capítulo. *Continúa en la página siguiente.*

Símbolo	Significado	Definición
$\Lambda_\alpha(k, t)$	Conjunto de valores de λ que verifican la primera restricción del OPC, fijadas k y t . Si existe λ_α , entonces $\Lambda_\alpha = [2, \lambda_\alpha]$; en otro caso, $\Lambda_\alpha = \emptyset$.	(4.119), p. 268
$\Lambda_\beta(k, t)$	Conjunto de valores de λ que verifican la segunda restricción del OPC, fijadas k y t ; es igual a $[\lambda_\beta, +\infty[$.	(4.124), p. 271
$\Lambda(k, t)$	Conjunto de valores de λ que cumplen las restricciones del problema OPC-NLP(k, t); es igual a $\Lambda_\alpha(k, t) \cap \Lambda_\beta(k, t)$.	(4.132), p. 275
$\lambda'(k, t)$	Si existe, solución óptima del problema OPC-NLP(k, t). Valor que minimiza el valor de f , fijados k y t y cumpliéndose todas las restricciones.	(4.131), p. 274
$\lambda_{\text{máx}}(k, t)$	Si existe, valor que maximiza el valor de f , fijados k y t .	Corol. 4.4, p. 243
(k^*, t^*, λ^*)	Si existe, solución óptima del problema OPC-sb. Valor que minimiza el valor de f , acotados superiormente k y t a $K_{\text{máx}}$ y $T_{\text{máx}}$, respectivamente, y cumpliéndose todas las restricciones.	(4.134), p. 279
iG	Estado del procedimiento de cualificación del precontrol que se produce tras observar i piezas verdes consecutivas.	Sec. 3.3.2, p. 114
jY	Estado del procedimiento de cualificación del precontrol que se produce tras observar j piezas amarillas consecutivas.	Sec. 3.3.2, p. 114
$P(x)$	Polinomio auxiliar.	(4.37), p. 229
$Q(x)$	Polinomio auxiliar.	(4.39), p. 230
$R(x)$	Función auxiliar.	(4.42), p. 232
$C^{(m)}(\lambda)$	Función auxiliar, igual a $R^{(m)}(p_g(\lambda))$.	(4.44), p. 234
$D^{(m)}(\lambda)$	Función auxiliar, igual a $R^{(m)}(p_y(\lambda))$.	(4.44), p. 234

Tabla 4.1: Relación y significado de las expresiones matemáticas particulares empleadas en el presente capítulo. *Continúa en la página siguiente.*

Símbolo	Significado	Definición
$v(\lambda)$	Función auxiliar, igual a $C^{(k)}(\lambda) - D^{(t)}(\lambda)$.	(4.71), p. 246
$G(k, \lambda C_p, \delta)$	Función auxiliar, igual a $\sum_{i=0}^{k-1} p_g(\lambda C_p, \delta)^i$.	(4.52), p. 238
$Y(t, \lambda C_p, \delta)$	Función auxiliar, igual a $\sum_{j=0}^{t-1} p_y(\lambda C_p, \delta)^j$.	(4.30), p. 224
$V(t, \lambda C_p, \delta)$	Función auxiliar, igual a $p_g(\lambda C_p, \delta) \sum_{j=0}^{t-1} p_y(\lambda C_p, \delta)^j$.	(4.18), p. 216

Tabla 4.1: Relación y significado de las expresiones matemáticas particulares empleadas en el presente capítulo.

Recordemos que en el mismo capítulo fue introducida la versión *acotada* del problema OPC (4.5), mediante la expresión (3.36), para hacer referencia al problema de precontrol que realmente tratan de resolver los algoritmos de optimización de propósito general. El así llamado problema OPC-b puede expresarse con los ajustes de notación introducidos en el presente capítulo como sigue:

$$\left. \begin{array}{l}
 \text{Min} \quad f(k, t, \lambda | C_p) \\
 \text{s.a:} \quad \alpha(k, t, \lambda | C_p) \leq \alpha^* \\
 \quad \quad \beta(k, t, \lambda | C_p, \delta^*) \leq \beta^* \\
 \quad \quad 1 \leq k \leq K_{\text{máx}} \\
 \quad \quad 1 \leq t \leq T_{\text{máx}} \\
 \quad \quad 2 \leq \lambda \leq L_{\text{máx}} \\
 \quad \quad k, t \in \mathbb{Z}, \lambda \in \mathbb{R}
 \end{array} \right\} . \quad (4.6)$$

Como se deducirá de los resultados que demostraremos en el presente capítulo (en concreto, del teorema 4.3), en este caso puede considerarse innecesario establecer de antemano una cota superior para el valor del parámetro λ . Por dicha razón, el problema de optimización objeto último de estudio y cuya resolución abordamos realmente en el capítulo 5 es el que llamaremos *problema OPC-sb* ($\alpha^*, \beta^*, \delta^*, K_{\text{máx}}, T_{\text{máx}}$), o sencillamente problema OPC-sb (por *semi-bounded*), y que definimos de la siguiente

forma:

$$\begin{array}{r}
 \text{Min} \quad f(k, t, \lambda \mid C_p) \\
 \text{s.a:} \quad \alpha(k, t, \lambda \mid C_p) \leq \alpha^* \\
 \quad \quad \beta(k, t, \lambda \mid C_p, \delta^*) \leq \beta^* \\
 \quad \quad \quad 1 \leq k \leq K_{\text{máx}} \\
 \quad \quad \quad 1 \leq t \leq T_{\text{máx}} \\
 \quad \quad \quad \lambda \geq 2 \\
 \quad \quad k, t \in \mathbb{Z}, \lambda \in \mathbb{R}
 \end{array} \left. \vphantom{\begin{array}{r} \text{Min} \\ \text{s.a:} \end{array}} \right\}, \quad (4.7)$$

en el cual los parámetros C_p , α^* , β^* , δ^* , $K_{\text{máx}}$ y $T_{\text{máx}}$ están definidos y delimitados de manera idéntica a como se especificó en la definición original del problema OPC-b (3.36).

Como puede observarse, los tres problemas, (4.5), (4.6) y (4.7), comparten la misma función objetivo y las mismas restricciones sobre las probabilidades de error α y β . Lo que los diferencia es la presencia o ausencia de cotas superiores para las variables k , t y λ .

4.1.2. Objetivos, justificación y estructura

Con el fin de clarificar y facilitar el seguimiento de los contenidos de este capítulo, se enuncian a continuación los dos grandes objetivos específicos que se persiguen:

1. **Mostrar que las probabilidades α y β son, de modo respectivo, estrictamente creciente y decreciente como función de λ , en todo su dominio, fijados el resto de parámetros (tanto las variables k y t como los parámetros C_p y, en el caso de β , también el parámetro δ).** Se demostrará que la probabilidad de que el precontrol termine cualificando el proceso p_Q , directamente relacionada con las funciones α y β , es estrictamente decreciente como función de λ , fijados el resto de parámetros (teorema 4.1). Como corolario se obtendrán los resultados buscados (corolarios 4.2 y 4.3).

Recordemos que el parámetro λ es, por su definición (2.24), inversamente proporcional a la anchura de la zona verde. Por tanto,

es lógico pensar que, a medida que la zona verde se estrecha (es decir, a medida que aumenta el valor de λ) y fijadas el resto de condiciones, la probabilidad p_Q de aceptar el proceso como correcto se reduce. Sin embargo, esto *no* es algo tan obvio de deducir de manera inmediata a partir de la definición de p_Q como pudiera parecer.

Efectivamente, teniendo en cuenta que p_Q representa la probabilidad de *observar k piezas verdes consecutivas sin antes observar t amarillas consecutivas ni ninguna roja* en un muestreo aleatorio secuencial sin un tamaño muestral limitado de antemano, una primera manera bastante natural de expresar p_Q es la siguiente:

$$p_Q = p_g^k \cdot \Pr(\text{no observar } k \text{ verdes consecutivas ni } t \text{ amarillas consecutivas} \\ \text{ni } 1 \text{ roja en la secuencia de extracciones previas a} \\ \text{las } k \text{ últimas, no siendo, además, verde la última pieza} \\ \text{de dicha secuencia.}) \quad (4.8)$$

Como también demostraremos —esto sí es claro—, la probabilidad de pieza verde p_g y de pieza amarilla p_y disminuye y aumenta, respectivamente, según se incrementa λ . En consecuencia, es evidente que el primer término de la expresión (4.8) disminuye con el aumento de λ . En cambio, el segundo término de (4.8) no está claro a priori si se incrementa o si disminuye cuando aumenta el valor de λ , ya que —razonando a grandes rasgos— cuando esto sucede, por un lado se hace más improbable *no observar t amarillas consecutivas* pero por otro lado se hace más probable *no observar k verdes consecutivas*, además de que no resulta trivial obtener una expresión concreta para dicho término².

²Precisamente, la expresión final para la probabilidad de cualificar el proceso (2.12), obtenida por San Matías et al. (2004) de forma recursiva, reproduce la estructura recogida en (4.8), lo cual permite deducir una expresión formal para el segundo término de (4.8), que sería la siguiente: $\sum_{j=0}^{t-1} p_y^j / \left(1 - \sum_{j=1}^{t-1} p_y^j \sum_{i=1}^{k-1} p_g^i\right)$. El denominador de esta última expresión contiene el producto de un término que aumenta con λ con otro que disminuye

Por supuesto, son posibles otras maneras de enfocar la demostración, pero, a pesar de lo intuitivo del resultado, ninguna de ellas parece conducir a una conclusión de manera rápida e inmediata.

En resumen, demostraremos que el comportamiento de las probabilidades α y β como funciones de λ es el esperado, lo cual acabamos de mostrar que no es tan sencillo como en un primer momento cabría pensar.

2. **Mostrar que la función objetivo f del problema OPC (4.5), considerada como función de λ y fijados arbitrariamente el resto de parámetros, no posee ningún mínimo local en todo el dominio de λ .** Se demostrará que $f(\lambda)$ es unimodal, en el sentido de que su gráfica posee una única *joroba* o *cresta* y ningún *valle* (teorema 4.2), y como corolario se obtendrá el resultado deseado (corolario 4.5).

La función objetivo del problema OPC representa, como sabemos, el número esperado de piezas a muestrear cuando el proceso está centrado. Tomada como función de λ (o, lo que es lo mismo, como función de la anchura de la zona verde W_G), y permaneciendo constantes el resto de condiciones, es relativamente lógico pensar que, a medida que la zona verde se estrecha (es decir, según λ aumenta), el número de piezas necesarias para terminar el procedimiento de precontrol (bien aceptando el proceso o bien rechazándolo) acabará por reducirse, cuando la aparición de un pieza verde que *rompa* una secuencia de amarillas sea cada vez más improbable, hasta tender asintóticamente a una constante no negativa (relacionada necesariamente con el número de amarillas consecutivas necesarias t y con la probabilidad de pieza roja); y también es razonable la idea de que, cuando λ se encuentre alrededor de un determinado valor intermedio (que correspondería a una zona verde no demasiado grande ni demasiado pequeña),

cuando λ aumenta de valor, lo cual pone de manifiesto lo que ya se intuía: que el segundo término de (4.8) *no tiene por qué* ser una función monótona decreciente respecto de λ . De hecho, graficando este cociente como función de λ , y fijados arbitrariamente el resto de parámetros, se demuestra que en general *no* lo es.

tanto la aparición de piezas verdes como de amarillas sean ambas relativamente probables, lo cual hará más difícil la observación de secuencias de k verdes o t amarillas consecutivas y provocará, por tanto, un incremento del tamaño muestral esperado para terminar el precontrol.

Estas hipótesis basadas en la intuición parecen confirmarse al realizar gráficos exploratorios que representan f como función de λ , dando valores arbitrarios al resto de parámetros involucrados (se mostrarán algunos ejemplos más adelante). En el presente capítulo se dará un salto cualitativo, pues demostraremos formalmente que este comportamiento descrito para $f(\lambda)$ es cierto, cualquiera que sea el valor de los parámetros fijados. Al igual que la anterior, la demostración no es trivial, como se verá.

Como ya se ha dicho, estos objetivos son pertinentes en tanto en cuanto son necesarios (y suficientes) para abordar el diseño de algoritmos exactos específicos para la resolución del problema del OPC, además de por el propio hecho en sí de aportar nuevo conocimiento sobre dicho problema.

El resto del capítulo se estructura como sigue. En la sección 4.2 se demuestran algunos resultados elementales acerca del comportamiento de las probabilidades de obtener una pieza verde y una amarilla, los cuales serán utilizados de manera auxiliar en demostraciones posteriores. En la sección 4.3, concretamente en la subsección 4.3.1, se abordan los dos grandes objetivos enunciados acerca del comportamiento de las tres funciones involucradas en el problema OPC, α , β y f , como funciones del parámetro λ , que se materializan en la demostración de los teoremas 4.1 y 4.2, respectivamente. El resto de la sección se dedica a probar otras propiedades que tendrán utilidad posterior, relacionadas con el comportamiento de α , β y f como funciones de k y t . En la sección 4.4 se enuncian y demuestran las consecuencias que tienen en la resolución del problema OPC los resultados probados en las secciones anteriores. En concreto, el teorema 4.3 proporciona la base formal para un método de optimización *local* del valor del parámetro λ , y el corolario 4.13 y

la proposición 4.10 justifican teóricamente la eficacia del método de resolución de problemas de OPC que propondremos en el capítulo 5. Finalmente, en la sección 4.5 se mostrará la validez de los resultados obtenidos en aquellos casos en que la característica de calidad no siga una distribución normal; y terminaremos resumiendo las principales aportaciones de este capítulo en la sección 4.6.

4.2. Algunos resultados básicos

En primer lugar demostraremos algunos resultados sobre el comportamiento de las probabilidades p_g y p_y como funciones del parámetro λ , ampliando lo que ya se ha visto en la sección 2.5.

Proposición 4.1 (Propiedades de p_g como función de λ). *Fijados el resto de parámetros, la probabilidad de obtener una observación verde p_g es continua, derivable y estrictamente decreciente como función de λ en todo el dominio de λ , $[2, +\infty[$. Además, se cumple:*

$$p_g(\lambda = 2) = \Phi(\delta + 3C_p) - \Phi(\delta - 3C_p) = S, \quad (4.9)$$

$$\lim_{\lambda \rightarrow +\infty} p_g(\lambda) = 0 \quad (4.10)$$

y

$$p_g'(\lambda) = -\frac{6C_p}{\lambda^2} \left(\varphi \left(\delta + \frac{6C_p}{\lambda} \right) + \varphi \left(\delta - \frac{6C_p}{\lambda} \right) \right), \quad (4.11)$$

donde Φ y φ representan, respectivamente, la función de distribución acumulada y la función de densidad de una variable aleatoria normal tipificada.

Demostración. Es una conclusión inmediata a partir de la expresión de p_g (2.21),

$$p_g(\lambda \mid C_p, \delta) = \Phi \left(\delta + \frac{6C_p}{\lambda} \right) - \Phi \left(\delta - \frac{6C_p}{\lambda} \right),$$

de la definición de S (4.1) y del hecho de que la función de distribución Φ de una variable aleatoria normal tipificada (véase (2.23)) es estrictamente creciente en toda la recta real (cualquier función de distribución

acumulada es, por definición, monótona creciente), además de continua y diferenciable.

La expresión para la derivada de $p_g(\lambda)$ (4.11) se obtiene aplicando la regla de la cadena a (2.21), teniendo en cuenta que la derivada de $\Phi(z)$ es

$$\Phi'(z) = \varphi(z), \quad (4.12)$$

la función de densidad de una variable aleatoria normal tipificada, cuya expresión es la siguiente:

$$\varphi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}, \quad -\infty < z < +\infty. \quad (4.13)$$

□

El resultado anterior es lógico, ya que, por la definición de λ (2.24), incrementar dicho parámetro equivale a reducir la anchura de la zona verde W_G . Por tanto, al incrementarse el valor de λ y fijados el resto de parámetros, la probabilidad de obtener una observación verde p_g debe disminuir.

De manera análoga a lo realizado con p_g , analizamos a continuación el comportamiento de la probabilidad de obtener una observación amarilla p_y en función del parámetro λ .

Corolario 4.1 (Propiedades de p_y como función de λ). *Fijados el resto de parámetros, la probabilidad de obtener una observación amarilla p_y es continua, diferenciable y estrictamente creciente como función de λ en todo el dominio de λ , $[2, +\infty[$. Además, se cumple:*

$$p_y(\lambda = 2) = 0, \quad (4.14)$$

$$\lim_{\lambda \rightarrow +\infty} p_y(\lambda) = \Phi(\delta + 3C_p) - \Phi(\delta - 3C_p) = S \quad (4.15)$$

y

$$p_y'(\lambda) = -p_g'(\lambda) \tag{4.16}$$

$$= \frac{6C_p}{\lambda^2} \left(\varphi \left(\delta + \frac{6C_p}{\lambda} \right) + \varphi \left(\delta - \frac{6C_p}{\lambda} \right) \right). \tag{4.17}$$

Demostración. Se puede concluir directamente a partir de la expresión de p_y (2.22),

$$p_y(\lambda | C_p, \delta) = \Phi(\delta + 3C_p) - \Phi(\delta - 3C_p) - p_g(\lambda | C_p, \delta),$$

de la definición de S (4.1) y de las propiedades vistas sobre p_g como función de λ probadas en la proposición 4.1. \square

Los resultados que acabamos de demostrar explican el comportamiento de las probabilidades de obtener una pieza verde y una amarilla, p_g y p_y , según se va estrechando la zona verde (es decir, según se incrementa el valor del parámetro λ), fijados los parámetros δ y C_p . Bajo estos supuestos, la suma de ambas probabilidades es constante e igual a S (4.1), la probabilidad de obtener una pieza dentro de especificaciones. Según la proposición 4.1 y el corolario 4.1, cuando λ toma el valor 2 (lo que equivale, de acuerdo con (2.24), a que la zona verde coincide con el rango de especificaciones, es decir, la zona amarilla *no existe*), la probabilidad p_g lógicamente coincide con S , mientras que p_y obviamente es igual a 0. A medida que aumenta λ (es decir, a medida que se reduce la anchura de la zona verde), p_g y p_y disminuyen y aumentan, respectivamente, a la misma *velocidad* ($p_y'(\lambda) = -p_g'(\lambda)$ según (4.16), consecuencia directa de que la suma de $p_g(\lambda)$ y $p_y(\lambda)$ es constante), hasta que, en el límite, cuando λ tiende a infinito, la zona verde tiende a tener anchura nula y por tanto p_g tiende a 0, mientras que p_y tiende asintóticamente a S .

La figura 4.1 muestra el comportamiento de p_g y p_y en función de λ , fijados los valores de los parámetros δ y C_p .

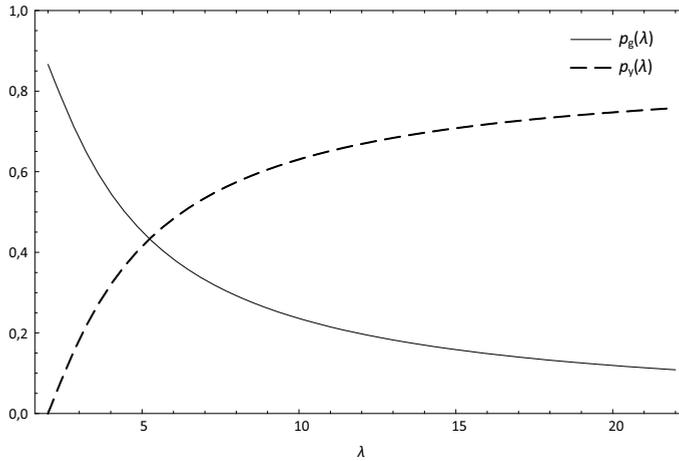


Figura 4.1: Probabilidades de muestrear una pieza verde y una amarilla como función del parámetro λ , en un proceso con desviación $\delta = 0$ y capacidad potencial $C_p = 0,5$.

4.3. Propiedades de las funciones del precontrol óptimo

A continuación demostramos algunas propiedades de las funciones f , α y β involucradas en el problema del precontrol óptimo (4.5), como funciones de los parámetros del precontrol k , t y λ . Estos resultados amplían el conocimiento que tenemos del problema OPC (4.5), lo cual hace viable el diseño de algoritmos específicos para su resolución.

4.3.1. Comportamiento como función de λ

En esta sección estudiamos el comportamiento de las funciones del modelo del precontrol óptimo α , β y f como funciones de la variable continua λ , fijado el valor de las variables k y t y el de los parámetros del proceso C_p y δ .

En primer lugar abordamos la demostración de una propiedad que intuitivamente parece cierta, pero cuya prueba no resulta trivial, como se verá: a medida que la zona verde del precontrol se estrecha (es decir, según va aumentando el valor de λ), y fijados el resto de parámetros, disminuye la probabilidad de cualificar el proceso.

Para probar dicha propiedad, utilizaremos el lema que demostramos a continuación.

Lema 4.1. *Fijados el resto de parámetros, la función*

$$V(\lambda) = p_g(\lambda) \sum_{j=0}^{t-1} p_y(\lambda)^j \tag{4.18}$$

es estrictamente decreciente en todo el dominio de λ $[2, +\infty[$, para todo valor entero $t \geq 1$.

Demostración. En esta demostración utilizaremos la expresión de la suma parcial de una serie geométrica de razón $r \neq 1$ (3.20)

$$\sum_{i=0}^{m-1} r^i = \frac{1 - r^m}{1 - r}.$$

Asimismo, usaremos los resultados demostrados en la proposición 4.1 y el corolario 4.1 sobre las probabilidades p_g y p_y y su derivada, como funciones de λ . Por último, haremos uso de la expresión S (4.1) como abreviatura de la suma $p_g + p_y$ (véase (4.2)) en determinados puntos de la demostración, con el fin de hacerla más clara y fácil de seguir.

Para probar que V es estrictamente decreciente como función de λ , obtendremos una expresión analítica para su derivada y demostraremos que ésta es negativa en todo el dominio de λ , $[2, +\infty[$.

Según (3.20) podemos expresar (4.18) de la manera siguiente:

$$V(\lambda) = p_g(\lambda) \frac{1 - p_y(\lambda)^t}{1 - p_y(\lambda)}. \tag{4.19}$$

Calculamos y desarrollamos la derivada de V con respecto a λ a partir de (4.19):

$$\begin{aligned} V'(\lambda) &= p_g'(\lambda) \frac{1 - p_y(\lambda)^t}{1 - p_y(\lambda)} \\ &\quad + p_g(\lambda) \frac{-tp_y(\lambda)^{t-1} \cdot p_y'(\lambda) (1 - p_y(\lambda)) - (1 - p_y(\lambda)^t) (-p_y'(\lambda))}{(1 - p_y(\lambda))^2} \\ &= \frac{1}{(1 - p_y(\lambda))^2} \cdot \left(p_g'(\lambda) (1 - p_y(\lambda)^t) (1 - p_y(\lambda)) \right. \\ &\quad \left. + p_g(\lambda) p_y'(\lambda) \left(-tp_y(\lambda)^{t-1} + tp_y(\lambda)^t + 1 - p_y(\lambda)^t \right) \right); \end{aligned}$$

aplicando la equivalencia (4.16) vista en el corolario 4.1, $p_y'(\lambda) = -p_g'(\lambda)$, se tiene:

$$\begin{aligned} V'(\lambda) &= \frac{p_g'(\lambda)}{(1 - p_y(\lambda))^2} \cdot \left((1 - p_y(\lambda)^t) (1 - p_y(\lambda)) \right. \\ &\quad \left. + p_g(\lambda) \left(-1 + tp_y(\lambda)^{t-1} - (t-1)p_y(\lambda)^t \right) \right) \\ &= \frac{p_g'(\lambda)}{(1 - p_y(\lambda))^2} \cdot \left(1 - p_y(\lambda) - p_y(\lambda)^t + p_y(\lambda)^{t+1} \right. \\ &\quad \left. - p_g(\lambda) + tp_g(\lambda)p_y(\lambda)^{t-1} - (t-1)p_g(\lambda)p_y(\lambda)^t \right); \end{aligned}$$

sabemos que $p_g(\lambda) + p_y(\lambda)$ es una cantidad que no depende de λ , a la que hemos llamado S (véase (4.2)):

$$\begin{aligned} V'(\lambda) &= \frac{p_g'(\lambda)}{(1 - p_y(\lambda))^2} \\ &\quad \cdot \left(1 - S + p_y(\lambda)^{t-1} \left(tp_g(\lambda) - (1 + (t-1)p_g(\lambda)) p_y(\lambda) + p_y(\lambda)^2 \right) \right); \end{aligned}$$

de nuevo sustituimos $p_y(\lambda)$ por $S - p_g(\lambda)$ y simplificamos:

$$\begin{aligned}
 V'(\lambda) &= \frac{p_g'(\lambda)}{(1 - p_y(\lambda))^2} \cdot \left(1 - S + p_y(\lambda)^{t-1} \left(t p_g(\lambda) \right. \right. \\
 &\quad \left. \left. - (1 + (t - 1)p_g(\lambda)) (S - p_g(\lambda)) + (S - p_g(\lambda))^2 \right) \right) \\
 &= \frac{p_g'(\lambda)}{(1 - p_y(\lambda))^2} \cdot \left(1 - S + p_y(\lambda)^{t-1} \left(-S + S^2 + (1 - S + t - tS)p_g(\lambda) \right. \right. \\
 &\quad \left. \left. + t p_g(\lambda)^2 \right) \right) \\
 &= \frac{p_g'(\lambda)}{(1 - p_y(\lambda))^2} \cdot \left(1 - S + t p_g(\lambda)^2 p_y(\lambda)^{t-1} \right. \\
 &\quad \left. + p_y(\lambda)^{t-1} \left(-S(1 - S) + (1 - S)p_g(\lambda) + t(1 - S)p_g(\lambda) \right) \right) \\
 &= \frac{p_g'(\lambda)}{(1 - p_y(\lambda))^2} \cdot \left(t p_g(\lambda)^2 p_y(\lambda)^{t-1} \right. \\
 &\quad \left. + (1 - S) \left(1 + (-S + (1 + t)p_g(\lambda)) p_y(\lambda)^{t-1} \right) \right). \tag{4.20}
 \end{aligned}$$

Dado que $(1 - p_y(\lambda))^2 > 0$ y que, por la proposición 4.1, $p_g'(\lambda) < 0$ para todo $\lambda \in [2, +\infty[$, a partir de la expresión (4.20) se deduce:

$$\begin{aligned}
 V'(\lambda) < 0 &\Leftrightarrow t p_g(\lambda)^2 p_y(\lambda)^{t-1} \\
 &\quad + (1 - S) \left(1 + (-S + (1 + t)p_g(\lambda)) p_y(\lambda)^{t-1} \right) > 0 \\
 &\Leftrightarrow 1 + (-S + (1 + t)p_g(\lambda)) p_y(\lambda)^{t-1} > \frac{-t p_g(\lambda)^2 p_y(\lambda)^{t-1}}{1 - S} \\
 &\Leftrightarrow 1 > p_y(\lambda)^{t-1} \left(S - (1 + t)p_g(\lambda) - \frac{t p_g(\lambda)^2}{1 - S} \right). \tag{4.21}
 \end{aligned}$$

Ahora bien, por un lado $t \geq 1$, $p_g(\lambda) > 0$ para todo $\lambda \in [2, +\infty[$ y

$0 < S < 1$ según (4.4), y por tanto

$$S - (1+t)p_g(\lambda) - \frac{tp_g(\lambda)^2}{1-S} < S < 1. \quad (4.22)$$

Por otro lado, $0 \leq p_y(\lambda) < 1$, y en consecuencia

$$0 \leq p_y(\lambda)^{t-1} < 1, \quad \forall \lambda \in [2, +\infty[. \quad (4.23)$$

De (4.22) y (4.23) se deduce que la expresión (4.21) es cierta, y que por tanto $V'(\lambda) < 0$ para todo $\lambda \in [2, +\infty[$, esto es, que $V(\lambda)$ es estrictamente decreciente como función de λ en todo su dominio, como se quería demostrar. \square

La figura 4.2 muestra el comportamiento de la función auxiliar $V(\lambda)$ para varios valores del parámetro t .

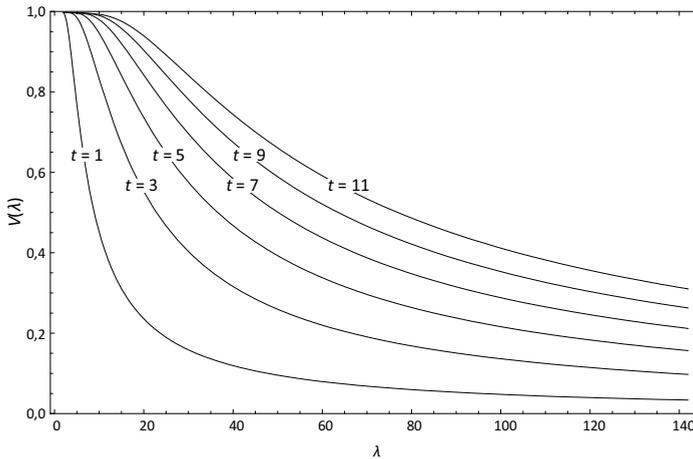


Figura 4.2: Comportamiento de la función $V(\lambda)$ para varios valores del parámetro t , en un proceso con desviación $\delta = 0$ y capacidad potencial $C_p = 1$.

A continuación abordamos la demostración del resultado que se ha enunciado antes de manera informal.

Teorema 4.1 (Comportamiento de la probabilidad de cualificar como función de λ). *Fijados el resto de parámetros, la probabilidad de cualificar el proceso p_Q es continua, diferenciable y estrictamente decreciente como función de λ en todo el dominio de λ , $[2, +\infty]$. Además, se cumple:*

$$p_Q(\lambda = 2) = p_g(\lambda = 2)^k = s^k, \quad (4.24)$$

$$\lim_{\lambda \rightarrow +\infty} p_Q(\lambda) = 0. \quad (4.25)$$

Demostración. Fijados arbitrariamente dentro de sus respectivos rangos los valores de los parámetros k , t , C_p y δ , se quiere probar que la función $p_Q(\lambda \mid k, t, C_p, \delta)$, que se define como

$$p_Q(\lambda \mid k, t, C_p, \delta) = p_Q(k, t, \lambda \mid C_p, \delta)$$

y que, según lo convenido en la subsección 4.1.1, nombraremos de manera abreviada como $p_Q(\lambda)$, es continua, diferenciable y, lo que es más importante, *estrictamente decreciente* como función de λ .

Consideraremos la expresión (2.12) para $p_Q(\lambda)$:

$$p_Q(\lambda) = p_g(\lambda)^k \frac{\sum_{j=0}^{t-1} p_y(\lambda)^j}{1 - \sum_{j=1}^{t-1} p_y(\lambda)^j \sum_{i=1}^{k-1} p_g(\lambda)^i}.$$

A partir de dicha expresión, y aplicando la proposición 4.1 y el corolario 4.1, es claro que $p_Q(\lambda)$ es continua y diferenciable en todo el dominio de λ , $[2, +\infty]$. El único punto donde podría fallar la continuidad es aquel para el cual se anulase el denominador, pero ya se vio en la sección 2.5 que la expresión (2.12) estaba bien definida, y que, en concreto, su denominador siempre era estrictamente mayor que cero³.

Una vez probado que $p_Q(\lambda)$ es continua, y apoyándonos en lo

³Existe también un problema formal con la continuidad en el punto $\lambda = 2$, ya que el término $\sum_{j=0}^{t-1} p_y(\lambda = 2)^j$ contiene una indeterminación del tipo 0^0 (para $j = 0$). Se trata de una discontinuidad evitable, dado que $\lim_{\lambda \rightarrow 2^+} p_y(\lambda)^0 = 1$, pudiendo por tanto

demostrado en la proposición 4.1 y en el corolario 4.1, es inmediato calcular $p_Q(2)$ y $\lim_{\lambda \rightarrow +\infty} p_Q(\lambda)$ a partir de (2.12), (2.14), (4.9), (4.10), (4.14) y (4.15):

$$\begin{aligned}
 p_Q(2) &= p_g(2)^k \frac{\sum_{j=0}^{t-1} p_y(2)^j}{1 - \sum_{j=1}^{t-1} p_y(2)^j \sum_{i=1}^{k-1} p_g(2)^i} \\
 &= S^k \frac{1}{1 - 0 \cdot \sum_{i=1}^{k-1} S^i} = S^k, \\
 \lim_{\lambda \rightarrow +\infty} p_Q(\lambda) &= \lim_{\lambda \rightarrow +\infty} p_g(\lambda)^k \lim_{\lambda \rightarrow +\infty} \left(\frac{\sum_{j=0}^{t-1} p_y(\lambda)^j}{1 - \sum_{j=1}^{t-1} p_y(\lambda)^j \sum_{i=1}^{k-1} p_g(\lambda)^i} \right) \\
 &= 0 \cdot \frac{\sum_{j=0}^{t-1} S^j}{1 - \sum_{j=1}^{t-1} S^j \cdot 0} = 0.
 \end{aligned}$$

Realizaremos la demostración de que $p_Q(\lambda)$ es decreciente por inducción sobre el valor del parámetro k , haciendo uso del lema 4.1 que acabamos de probar.

Dado un valor k_0 de k , en esta demostración denotaremos de manera abreviada la función $p_Q(\lambda \mid k = k_0)$ como $p_Q^{(k_0)}(\lambda)$.

Para $k = 1$, tenemos según (2.12):

$$p_Q^{(1)}(\lambda) = p_g(\lambda) \sum_{j=0}^{t-1} p_y(\lambda)^j, \tag{4.26}$$

asumir sin reparos $\sum_{j=0}^{t-1} p_y(\lambda = 2)^j = 1, \forall t \geq 1$. Por otro lado, al aplicar la fórmula para el cálculo de la suma parcial de una serie geométrica (3.20) al caso en que la razón toma el valor 0 se obtiene idéntico resultado.

que coincide con la expresión $V(\lambda)$ (véase (4.18)), y por tanto es estrictamente decreciente como función de λ en virtud del lema 4.1.

Para $k = 2$, desarrollamos la expresión de $p_Q^{(2)}(\lambda)$ y la ponemos en relación con $p_Q^{(1)}(\lambda)$:

$$\begin{aligned} \text{por (2.12):} \quad p_Q^{(2)}(\lambda) &= p_g(\lambda)^2 \frac{\sum_{j=0}^{t-1} p_y(\lambda)^j}{1 - p_g(\lambda) \sum_{j=1}^{t-1} p_y(\lambda)^j} \\ \text{por (4.26):} \quad &= p_Q^{(1)}(\lambda) \frac{p_g(\lambda)}{1 + p_g(\lambda) - p_Q^{(1)}(\lambda)}. \end{aligned} \quad (4.27)$$

Queremos demostrar que $p_Q^{(2)}(\lambda)$ es estrictamente decreciente; lo haremos calculando su derivada con respecto a λ y comprobando que ésta siempre es negativa. Partiendo de (4.27):

$$\begin{aligned} p_Q^{(2)'}(\lambda) &= p_Q^{(1)'}(\lambda) \frac{p_g(\lambda)}{1 + p_g(\lambda) - p_Q^{(1)}(\lambda)} \\ &\quad + p_Q^{(1)}(\lambda) \frac{p_g'(\lambda) (1 + p_g(\lambda) - p_Q^{(1)}(\lambda)) - p_g(\lambda) (p_g'(\lambda) - p_Q^{(1)'}(\lambda))}{(1 + p_g(\lambda) - p_Q^{(1)}(\lambda))^2} \\ &= \frac{p_g(\lambda) p_Q^{(1)'}(\lambda) + p_g(\lambda)^2 p_Q^{(1)'}(\lambda) + p_g'(\lambda) p_Q^{(1)}(\lambda) - p_g'(\lambda) p_Q^{(1)}(\lambda)^2}{(1 + p_g(\lambda) - p_Q^{(1)}(\lambda))^2} \\ &= \frac{p_Q^{(1)'}(\lambda) (p_g(\lambda) + p_g(\lambda)^2) + p_g'(\lambda) p_Q^{(1)}(\lambda) (1 - p_Q^{(1)}(\lambda))}{(1 + p_g(\lambda) - p_Q^{(1)}(\lambda))^2}. \end{aligned} \quad (4.28)$$

Hemos probado antes que $p_Q^{(1)}$ es estrictamente decreciente como función de λ , y por tanto su derivada $p_Q^{(1)'}(\lambda)$ es negativa para todo valor de λ en $[2, +\infty[$. Además, el término $p_g(\lambda) + p_g(\lambda)^2$ es siempre positivo, ya que lo son los sumandos que lo componen. Por otro lado, $p_g'(\lambda) < 0$

para todo $\lambda \in [2, +\infty[$, como se ha demostrado en la proposición 4.1. Y por último, necesariamente $p_Q^{(1)}(\lambda) \left(1 - p_Q^{(1)}(\lambda)\right) > 0$, ya que la función p_Q representa una probabilidad (la de que el procedimiento de precontrol termine cualificando el proceso), y por su definición cumple $0 < p_Q^{(k)}(\lambda) < 1$ para todo $k \geq 1$ y para todo $\lambda \in [2, +\infty[$.

Todo ello permite concluir que la expresión (4.28) es necesariamente negativa, para cualquier valor de λ en el intervalo $[2, +\infty[$, y con esto queda probado que $p_Q^{(k)}$ es estrictamente decreciente en dicho intervalo para $k = 2$.

Ahora, supongamos que $p_Q^{(k)}$ es estrictamente decreciente como función de λ , para un determinado valor de k . A continuación veremos cómo, partiendo de esta suposición, es posible llegar a demostrar que $p_Q^{(k+1)}(\lambda)$ es también estrictamente decreciente.

En primer lugar, expresamos $p_Q^{(k+1)}(\lambda)$ en relación a $p_Q^{(k)}(\lambda)$. Por (2.12):

$$\begin{aligned}
 p_Q^{(k+1)}(\lambda) &= p_g(\lambda)^{k+1} \frac{\sum_{j=0}^{t-1} p_y(\lambda)^j}{1 - \sum_{j=1}^{t-1} p_y(\lambda)^j \sum_{i=1}^k p_g(\lambda)^i} \\
 &= p_g(\lambda) \frac{p_g(\lambda)^k \sum_{j=0}^{t-1} p_y(\lambda)^j / 1 - \sum_{j=1}^{t-1} p_y(\lambda)^j \sum_{i=1}^{k-1} p_g(\lambda)^i}{1 - \sum_{j=1}^{t-1} p_y(\lambda)^j \sum_{i=1}^k p_g(\lambda)^i / 1 - \sum_{j=1}^{t-1} p_y(\lambda)^j \sum_{i=1}^{k-1} p_g(\lambda)^i} \\
 &= p_g(\lambda) \frac{p_Q^{(k)}(\lambda)}{1 - \sum_{j=1}^{t-1} p_y(\lambda)^j \sum_{i=1}^{k-1} p_g(\lambda)^i - p_g(\lambda)^k \sum_{j=1}^{t-1} p_y(\lambda)^j} \\
 &\qquad\qquad\qquad 1 - \sum_{j=1}^{t-1} p_y(\lambda)^j \sum_{i=1}^{k-1} p_g(\lambda)^i
 \end{aligned}$$

$$\begin{aligned}
 &= p_g(\lambda) \frac{p_Q^{(k)}(\lambda)}{1 - \frac{p_g(\lambda)^k \left(\sum_{j=1}^{t-1} p_y(\lambda)^j + 1 - 1 \right)}{1 - \sum_{j=1}^{t-1} p_y(\lambda)^j \sum_{i=1}^{k-1} p_g(\lambda)^i}} \\
 &= p_g(\lambda) \frac{p_Q^{(k)}(\lambda)}{1 - p_Q^{(k)}(\lambda) + \frac{p_g(\lambda)^k}{1 - \sum_{j=1}^{t-1} p_y(\lambda)^j \sum_{i=1}^{k-1} p_g(\lambda)^i}} \\
 &= p_g(\lambda) \frac{p_Q^{(k)}(\lambda)}{1 - p_Q^{(k)}(\lambda) + \frac{p_Q^{(k)}(\lambda)}{\sum_{j=0}^{t-1} p_y(\lambda)^j}}.
 \end{aligned}$$

Hemos obtenido la siguiente relación entre $p_Q^{(k+1)}$ y $p_Q^{(k)}$:

$$p_Q^{(k+1)}(\lambda) = p_Q^{(k)}(\lambda) \frac{p_g(\lambda)}{1 - p_Q^{(k)}(\lambda) + \frac{p_Q^{(k)}(\lambda)}{Y(\lambda)}}, \quad (4.29)$$

donde

$$Y(\lambda) = \sum_{j=0}^{t-1} p_y(\lambda)^j. \quad (4.30)$$

Es decir, según (4.18),

$$V(\lambda) = p_g(\lambda)Y(\lambda). \quad (4.31)$$

Nótese que, para $k = 1$, la fórmula (4.29) coincide con la expresión (4.27) obtenida para el caso particular.

Calculamos la derivada de $p_Q^{(k+1)}(\lambda)$ a partir de (4.29):

$$\begin{aligned}
p_Q^{(k+1)'}(\lambda) &= p_Q^{(k)'}(\lambda) \frac{p_g(\lambda)}{1 - p_Q^{(k)}(\lambda) + \frac{p_Q^{(k)}(\lambda)}{Y(\lambda)}} \\
&\quad + p_Q^{(k)}(\lambda) \left(p_g'(\lambda) \left(1 - p_Q^{(k)}(\lambda) + \frac{p_Q^{(k)}(\lambda)}{Y(\lambda)} \right) \right. \\
&\quad \left. - p_g(\lambda) \left(-p_Q^{(k)'}(\lambda) + \frac{p_Q^{(k)'}(\lambda)Y(\lambda) - p_Q^{(k)}(\lambda)Y'(\lambda)}{Y(\lambda)^2} \right) \right) \\
&\quad \cdot \frac{1}{\left(1 - p_Q^{(k)}(\lambda) + \frac{p_Q^{(k)}(\lambda)}{Y(\lambda)} \right)^2} \\
&= \frac{1}{\left(1 - p_Q^{(k)}(\lambda) + \frac{p_Q^{(k)}(\lambda)}{Y(\lambda)} \right)^2} \left(p_g(\lambda)p_Q^{(k)'}(\lambda) \left(1 - p_Q^{(k)}(\lambda) + \frac{p_Q^{(k)}(\lambda)}{Y(\lambda)} \right) \right. \\
&\quad + p_g'(\lambda)p_Q^{(k)}(\lambda) - p_g'(\lambda)p_Q^{(k)}(\lambda)^2 + \frac{p_g'(\lambda)p_Q^{(k)}(\lambda)^2}{Y(\lambda)} \\
&\quad \left. + p_g(\lambda)p_Q^{(k)}(\lambda)p_Q^{(k)'}(\lambda) - \frac{p_g(\lambda)p_Q^{(k)}(\lambda)p_Q^{(k)'}(\lambda)}{Y(\lambda)} + \frac{p_g(\lambda)p_Q^{(k)}(\lambda)^2Y'(\lambda)}{Y(\lambda)^2} \right) \\
&= \frac{1}{\left(1 - p_Q^{(k)}(\lambda) + \frac{p_Q^{(k)}(\lambda)}{Y(\lambda)} \right)^2} \left(p_g(\lambda)p_Q^{(k)'}(\lambda) + p_g'(\lambda)p_Q^{(k)}(\lambda) \right. \\
&\quad \left. - p_g'(\lambda)p_Q^{(k)}(\lambda)^2 + \frac{p_g'(\lambda)p_Q^{(k)}(\lambda)^2}{Y(\lambda)} + \frac{p_g(\lambda)p_Q^{(k)}(\lambda)^2Y'(\lambda)}{Y(\lambda)^2} \right) \\
&= \frac{1}{\left(1 - p_Q^{(k)}(\lambda) + \frac{p_Q^{(k)}(\lambda)}{Y(\lambda)} \right)^2} \left(p_g(\lambda)p_Q^{(k)'}(\lambda) \right. \\
&\quad + p_g'(\lambda)p_Q^{(k)}(\lambda) \left(1 - p_Q^{(k)}(\lambda) \right) \\
&\quad \left. + \frac{p_Q^{(k)}(\lambda)^2}{Y(\lambda)} \left(p_g'(\lambda) + \frac{p_g(\lambda)Y'(\lambda)}{Y(\lambda)} \right) \right). \tag{4.32}
\end{aligned}$$

Estamos suponiendo cierto que $p_Q^{(k)}$ es estrictamente decreciente como función de λ en todo su dominio, y por tanto $p_Q^{(k)'}(\lambda) < 0$. Además, $p_g(\lambda) > 0$ según (2.25). Por otro lado, $p_g'(\lambda) < 0$ por la proposición 4.1, y como ya se ha razonado antes para el caso $k = 2$, el término $p_Q^{(k)}(\lambda) (1 - p_Q^{(k)}(\lambda))$ es necesariamente positivo, así como también lo es $Y(\lambda)$ por definición (véase (4.30)). Por último, nótese que, a partir de (4.31),

$$V'(\lambda) = p_g'(\lambda)Y(\lambda) + p_g(\lambda)Y'(\lambda);$$

y dado que, como acabamos de mencionar, $Y(\lambda) > 0$ y, por el lema 4.1, $V'(\lambda) < 0$, tenemos que

$$p_g'(\lambda) + \frac{p_g(\lambda)Y'(\lambda)}{Y(\lambda)} = \frac{V'(\lambda)}{Y(\lambda)} < 0.$$

Todo esto permite concluir que la expresión (4.32), correspondiente a $p_Q^{(k+1)'}$ (λ), tiene signo negativo, cualquiera que sea el valor de λ en $[2, +\infty[$.

En resumen, partiendo de que $p_Q^{(k)}$ es estrictamente decreciente como función de λ hemos llegado a probar que también lo es $p_Q^{(k+1)}$, para un valor entero arbitrario de $k \geq 1$. También lo habíamos demostrado para los casos $k = 1$ y $k = 2$ en particular.

Por inducción, podemos afirmar que la tesis es cierta para todo valor de $k \geq 1$, esto es, que la probabilidad de cualificar el proceso p_Q es estrictamente decreciente como función de λ , fijados arbitrariamente el resto de parámetros k, t, C_p y δ , como se quería demostrar. \square

Del teorema anterior se deducen directamente las dos siguientes proposiciones.

Corolario 4.2 (Comportamiento de α como función de λ). *Fijados el resto de parámetros, la probabilidad de falsa alarma α es continua, diferenciable y estrictamente creciente como función de λ en todo el dominio de λ , $[2, +\infty[$.*

Además, se cumple:

$$\alpha(\lambda = 2) = 1 - p_g(\lambda = 2 \mid \delta = 0)^k = 1 - S(\delta = 0)^k \quad y \quad (4.33)$$

$$\lim_{\lambda \rightarrow +\infty} \alpha(\lambda) = 1. \quad (4.34)$$

Demostración. Es consecuencia inmediata del teorema 4.1 y del hecho de que, según (2.20), $\alpha(k, t, \lambda \mid C_p) = 1 - p_Q(k, t, \lambda \mid C_p, \delta = 0)$. \square

Corolario 4.3 (Comportamiento de β como función de λ). *Fijados el resto de parámetros, la probabilidad β de no detectar una determinada desviación δ del proceso es continua, diferenciable y estrictamente decreciente como función de λ en todo el dominio de λ , $[2, +\infty[$. Además, se cumple:*

$$\beta(\lambda = 2 \mid \delta) = p_g(\lambda = 2 \mid \delta)^k = S(\delta)^k \quad y \quad (4.35)$$

$$\lim_{\lambda \rightarrow +\infty} \beta(\lambda \mid \delta) = 0. \quad (4.36)$$

Demostración. Es también consecuencia inmediata del teorema 4.1 y del hecho de que $\beta(k, t, \lambda \mid C_p, \delta) = p_Q(k, t, \lambda \mid C_p, \delta)$, de acuerdo con (2.27). \square

Las figuras 4.3 y 4.4 ilustran, respectivamente, el comportamiento de α y β como función de λ descrito en estos corolarios.

A continuación abordamos el estudio del comportamiento de la función objetivo del precontrol óptimo f como función de λ , fijadas las variables k y t y el resto de parámetros.

En concreto, demostraremos que $f(\lambda)$ no tiene en $]2, +\infty[$ un mínimo (es decir, su representación gráfica no contiene ningún *valle*) y que, de hecho (excepto cuando k o t valen 1), posee un único máximo (global) en $[2, +\infty[$. En definitiva, probaremos que se trata de una función *unimodal*.

Tal y como apunta Brent (1973), el concepto de unimodalidad admite diferentes definiciones, algunas incluso opuestas entre sí. Aquí tomaremos

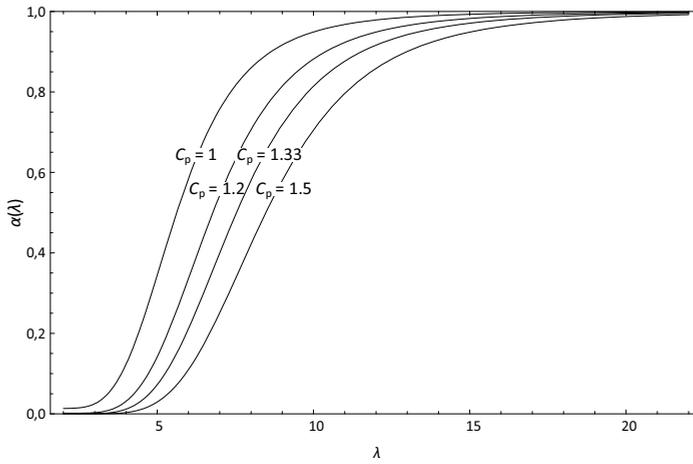


Figura 4.3: Comportamiento de α como función del parámetro λ , en el precontrol con $k = 5$ y $t = 2$, para varios valores del parámetro C_p .

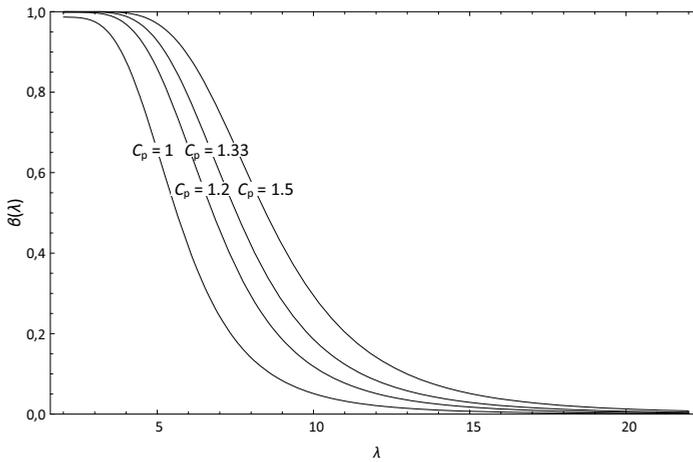


Figura 4.4: Comportamiento de $\beta(\cdot | \delta = 0)$ como función del parámetro λ , en el precontrol con $k = 5$ y $t = 2$, para varios valores del parámetro C_p .

como caracterización de unimodalidad una adaptación de la expresada por Brent⁴, ya que el comportamiento de $f(\lambda)$ se ajusta a ella, como demostraremos.

Así, diremos que una función F real de variable real continua y diferenciable es *unimodal* cuando, si existe un determinado x_1 en el dominio de F tal que $F'(x_1) < 0$, y dado $x_2 > x_1$, se cumple también que $F'(x_2) < 0$; en otras palabras, si cuando en un punto determinado la función F decrece, más adelante continúa decreciendo. Tal y como está enunciada, dentro de esta caracterización de *unimodalidad* tienen cabida también algunos casos especiales, como aquellos en los que la función F sea constante o monótona creciente (nunca decrece), o bien estrictamente decreciente (siempre decrece) en todo su dominio.

Esta propiedad es la que justamente probaremos que se cumple para nuestra función $f(\lambda)$. Para ello haremos uso de tres lemas previos que demostramos a continuación.

Lema 4.2. *Para cualquier valor $m \geq 1$, el polinomio*

$$P(x) = 1 - (m + 1)x^m + mx^{m+1} \quad (4.37)$$

es estrictamente decreciente y toma valores positivos cuando $x \in]0, 1[$. Además, $P(0) = 1$ y $P(1) = 0$.

Demostración. Es inmediato comprobar que $P(0) = 1$ y $P(1) = 0$:

$$\begin{aligned} P(0) &= 1 - (m + 1)0^m + m0^{m+1} = 1 - 0 + 0 = 1 \quad \text{y} \\ P(1) &= 1 - (m + 1)1^m + m1^{m+1} = 1 - m - 1 + m = 0. \end{aligned}$$

Calculamos la derivada de $P(x)$ a partir de su expresión (4.37):

$$\begin{aligned} P'(x) &= -m(m + 1)x^{m-1} + m(m + 1)x^m \\ &= -m(m + 1)x^{m-1}(1 - x). \end{aligned} \quad (4.38)$$

⁴Al contrario que aquí, Brent considera *unimodales* aquellas funciones que, gráficamente, tienen un único *valle* y ninguna *cresta*; además, maneja una definición de unimodalidad sin uso de derivadas, aplicable a funciones que ni siquiera tienen por qué ser continuas.

La expresión (4.38) es claramente negativa cuando $x \in]0, 1[$. Por tanto, queda demostrado que en dicho intervalo el polinomio P es estrictamente decreciente.

Además, dado que por ser un polinomio P es una función continua, y como P es decreciente en $]0, 1[$, podemos afirmar que para todo $x \in]0, 1[$

$$P(x) > P(1) = 0,$$

es decir, $P(x)$ toma valores estrictamente positivos cuando x está en el intervalo $]0, 1[$ (y también cuando $x = 0$), como se quería demostrar. \square

La figura 4.5 muestra la gráfica del polinomio $P(x)$ para varios valores de m .

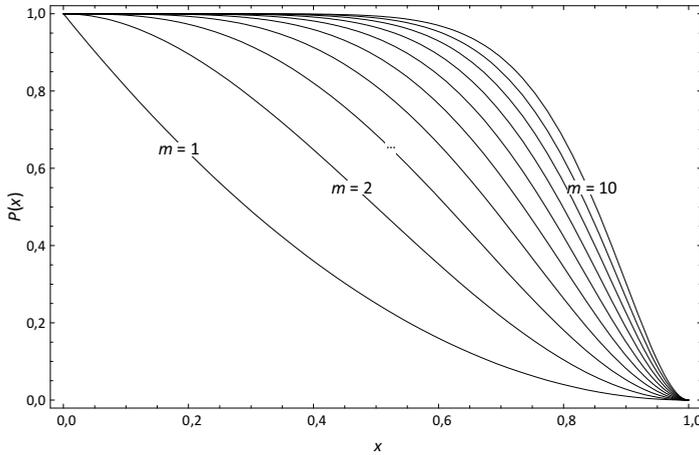


Figura 4.5: Comportamiento del polinomio auxiliar $P(x)$ entre 0 y 1, para varios valores del parámetro m .

Lema 4.3. Para cualquier valor entero $m \geq 1$, el polinomio

$$Q(x) = (m - 1) - (m + 1)x + (m + 1)x^m - (m - 1)x^{m+1} \quad (4.39)$$

toma valores no negativos cuando $x \in [0, 1]$.

Demostración. Lo demostraremos por inducción sobre m . Denotaremos por $Q^{(m_0)}$ al polinomio Q (4.39), fijado el valor de m a un valor concreto m_0 .

Para $m = 1$, es inmediato comprobar que Q es constante igual a cero:

$$Q^{(1)}(x) = 0 - 2x + 2x^1 - 0x^0 = -2x + 2x = 0,$$

para todo $x \in [0, 1]$.

Para $m = 2$,

$$Q^{(2)}(x) = 1 - 3x + 3x^2 - x^3 = (1 - x)^3,$$

que es mayor o igual a cero cuando x toma valores en el intervalo $[0, 1]$.

Supongamos que el resultado es cierto para un valor cualquiera de m ; es decir:

$$Q^{(m)}(x) = (m - 1) - (m + 1)x + (m + 1)x^m - (m - 1)x^{m+1} \geq 0 \quad (4.40)$$

para todo $x \in [0, 1]$.

Ahora, veamos si el resultado es entonces cierto para $m + 1$:

$$\begin{aligned} \text{por (4.39): } Q^{(m+1)}(x) &= m - (m + 2)x + (m + 2)x^{m+1} - mx^{m+2} \\ &= ((m - 1) - (m + 1)x + (m + 1)x^m - (m - 1)x^{m+1}) \\ &\quad + (1 - x - (m + 1)x^m + (2m + 1)x^{m+1} - mx^{m+2}) \\ \text{por (4.39): } &= Q^{(m)}(x) + (1 - (m + 1)x^m + mx^{m+1}) \\ &\quad - (x - (m + 1)x^{m+1} + mx^{m+2}) \\ &= Q^{(m)}(x) + (1 - x)(1 - (m + 1)x^m + mx^{m+1}). \end{aligned} \quad (4.41)$$

Si $x \in [0, 1]$, sabemos que $Q^{(m)}(x) \geq 0$ por hipótesis de inducción (4.40), que $1 - x \geq 0$ y que $1 - (m + 1)x^m + mx^{m+1} \geq 0$ por el lema 4.2. Con todo ello, podemos concluir que la expresión (4.41) es mayor o igual a cero, para todo $x \in [0, 1]$.

Partiendo de la suposición de que $Q^{(m)} \geq 0$ en $[0, 1]$ hemos deducido que el resultado también es cierto para $Q^{(m+1)}$; y también lo hemos probado para los casos particulares $m = 1$ y $m = 2$.

Razonando por inducción llegamos a la conclusión de que $Q(x) \geq 0$ cuando $x \in [0, 1]$, para cualquier valor entero positivo del parámetro m , como se quería demostrar. \square

La figura 4.6 ilustra el comportamiento del polinomio $Q(x)$ que acabamos de demostrar.

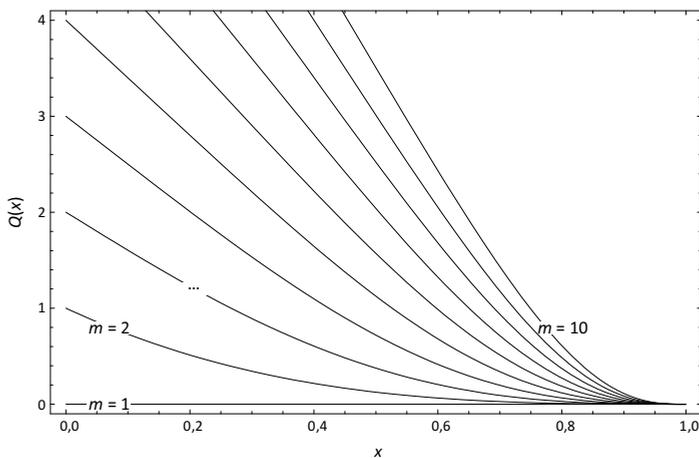


Figura 4.6: Comportamiento del polinomio auxiliar $Q(x)$ entre 0 y 1, para varios valores del parámetro m .

Lema 4.4. Para cualquier valor entero $m \geq 2$, la función

$$R^{(m)}(x) = \frac{1 - mx^{m-1} + (m-1)x^m}{(1-x^m)^2} \quad (4.42)$$

es estrictamente decreciente en $[0, 1]$; cuando $m = 1$, $R^{(1)}(x)$ es constante igual a cero.

Demostración. Calcularemos la derivada de $R^{(m)}(x)$ y estudiaremos su signo.

Para $m = 1$, tenemos

$$R^{(1)}(x) = \frac{1 - x^0 + 0x}{(1 - x)^2} = 0,$$

y por tanto $R^{(1)'}(x) = 0$ para todo $x \in]0, 1[$ ⁵.

Para $m \geq 2$:

$$\begin{aligned} R^{(m)'}(x) &= \frac{1}{(1 - x^m)^4} \left((-m(m-1)x^{m-2} + m(m-1)x^{m-1})(1 - x^m)^2 \right. \\ &\quad \left. + (1 - mx^{m-1} + (m-1)x^m) \cdot 2(1 - x^m)mx^{m-1} \right) \\ &= \frac{mx^{m-2}}{(1 - x^m)^3} \left((-(m-1) + (m-1)x)(1 - x^m) \right. \\ &\quad \left. + (1 - mx^{m-1} + (m-1)x^m) \cdot 2x \right) \\ &= \frac{mx^{m-2}}{(1 - x^m)^3} \left(-(m-1) + (m-1)x + (m-1)x^m \right. \\ &\quad \left. - (m-1)x^{m+1} + 2x - 2mx^m + 2(m-1)x^{m+1} \right) \\ &= \frac{mx^{m-2} (-(m-1) + (m+1)x - (m+1)x^m + (m-1)x^{m+1})}{(1 - x^m)^3}; \end{aligned}$$

por (4.39):

$$R^{(m)'}(x) = \frac{-mx^{m-2}Q(x)}{(1 - x^m)^3}. \quad (4.43)$$

Para $x \in]0, 1[$ y $m \geq 2$, es claro que el denominador de (4.43) $(1 - x^m)^3$ es positivo y que $mx^{m-2} > 0$, mientras que $Q(x) \geq 0$ por el lema 4.3.

Por tanto, puede concluirse que la expresión para la derivada de $R^{(m)}(x)$ (4.43) es negativa, cuando x varía en $]0, 1[$. Es decir, $R^{(m)}$ es una función estrictamente decreciente en $]0, 1[$, como se quería demostrar⁶. \square

⁵En realidad, en el caso $m = 1$ al calcular $R^{(1)}(0)$ se genera una indeterminación del tipo 0^0 . Se trata de una discontinuidad evitable, ya que al ser $R^{(1)}(x)$ constante e igual a cero en todo el intervalo $]0, 1[$, podemos asumir $R^{(1)}(0) = \lim_{x \rightarrow 0^+} R^{(1)}(x) = 0$.

⁶Por ser una función continua en $]0, 1[$, no hace falta exigir que la derivada de $R^{(m)}$

En la figura 4.7 se grafica la función $R^{(m)}(x)$ para varios valores de m .

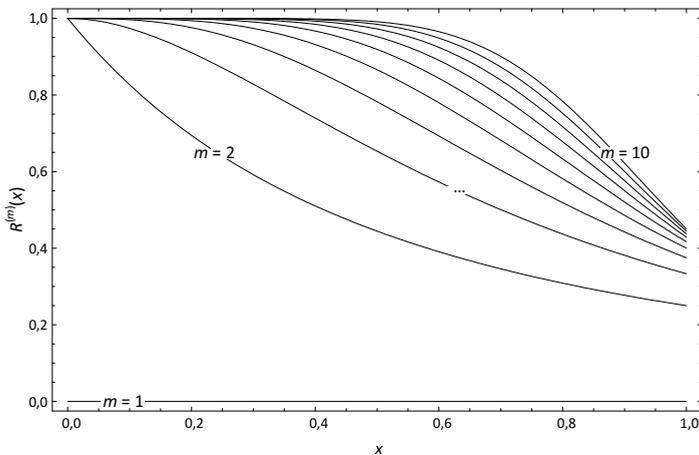


Figura 4.7: Comportamiento de la función auxiliar $R(x)$ entre 0 y 1, para varios valores del parámetro m .

Lema 4.5. Para cualquier valor entero $m \geq 2$, las funciones

$$C^{(m)}(\lambda) = R^{(m)}(p_g(\lambda)) \quad y \quad D^{(m)}(\lambda) = R^{(m)}(p_y(\lambda)) , \quad (4.44)$$

con $\lambda \in [2, +\infty[$, donde $R^{(m)}(x)$ es la función definida en el lema 4.4 según la expresión (4.42)

$$R^{(m)}(x) = \frac{1 - mx^{m-1} + (m-1)x^m}{(1-x^m)^2} ,$$

son continuas y, de modo respectivo, estrictamente creciente y estrictamente decreciente en $[2, +\infty[$.

en $x = 0$ sea negativa para afirmar que $R^{(m)}$ es estrictamente decreciente en $[0, 1[$, si lo es en $]0, 1[$.

Además, se cumple

$$C^{(m)}(2) = R^{(m)}(S) = \frac{1 - mS^{m-1} + (m-1)S^m}{(1 - S^m)^2}, \quad (4.45)$$

$$\lim_{\lambda \rightarrow +\infty} C^{(m)}(\lambda) = R^{(m)}(0) = 1, \quad (4.46)$$

$$D^{(m)}(2) = R^{(m)}(0) = 1, \quad (4.47)$$

$$\lim_{\lambda \rightarrow +\infty} D^{(m)}(\lambda) = R^{(m)}(S) = \frac{1 - mS^{m-1} + (m-1)S^m}{(1 - S^m)^2}. \quad (4.48)$$

Cuando $m = 1$, las funciones $C^{(1)}(\lambda)$ y $D^{(1)}(\lambda)$ son ambas constantes e iguales a cero.

Demostración. Tanto $C^{(m)}$ como $D^{(m)}$ están bien definidas en $[2, +\infty[$, ya que, según lo visto en la proposición 4.1 y el corolario 4.1, cuando λ varía en $[2, +\infty[$ entonces $p_g(\lambda)$ varía en $]0, S]$ mientras que $p_y(\lambda)$ lo hace en $[0, S[$, y dado que $S < 1$ (véase (4.4)), ambos intervalos quedan incluidos dentro de $[0, 1[$, dominio sobre el que se ha definido $R^{(m)}$ en el lema 4.4.

Asimismo, la continuidad de $R^{(m)}(x)$ (es un cociente de polinomios y el denominador no se anula en su dominio) y de $p_g(\lambda)$ y $p_y(\lambda)$ (demostradas éstas en la proposición 4.1 y el corolario 4.1, respectivamente) implican la continuidad tanto de $C^{(m)}(\lambda)$ como de $D^{(m)}(\lambda)$.

En el caso $m = 1$, trivialmente $C^{(1)}(\lambda) = D^{(1)}(\lambda) = 0$, para todo $\lambda \in [2, +\infty[$, como consecuencia de que, de acuerdo con el lema 4.4, $R^{(1)}(x) = 0$ para todo $x \in [0, 1[$.

Supongamos ahora que $m \geq 2$.

Según el lema 4.4 y la proposición 4.1, tanto $R^{(m)}(x)$ como $p_g(\lambda)$ son estrictamente decrecientes en sus respectivos dominios. Esto implica que la función $C^{(m)}(\lambda)$ es estrictamente creciente en $[2, +\infty[$; en efecto:

$$C^{(m)'}(\lambda) = (R^{(m)} \circ p_g)'(\lambda) = p_g'(\lambda)R^{(m)'}(p_g(\lambda)),$$

y ese producto es mayor que cero, para todo λ en $[2, +\infty[$, ya que ambos factores son menores que cero.

De manera análoga se puede comprobar que $D^{(m)}$ es estrictamente

decreciente como función de λ en todo su dominio. El lema 4.4 nos asegura que $R^{(m)}(x)$ es estrictamente decreciente, y el corolario 4.1 que $p_y(\lambda)$ es estrictamente creciente; junto con la equivalencia (4.16) $p_y'(\lambda) = -p_g'(\lambda)$, tenemos:

$$D^{(m)' }(\lambda) = (R^{(m)} \circ p_y)'(\lambda) = -p_g'(\lambda)R^{(m)' }(p_y(\lambda)) > 0, \quad \forall \lambda \in]2, +\infty[.$$

Por último, evaluamos $C^{(m)}$ y $D^{(m)}$ en los extremos de sus respectivos dominios. Aplicando que, según se vio en la proposición 4.1 y el corolario 4.1, $p_g(\lambda = 2) = S$ (4.9), $\lim_{\lambda \rightarrow +\infty} p_g(\lambda) = 0$ (4.10), $p_y(\lambda = 2) = 0$ (4.14) y $\lim_{\lambda \rightarrow +\infty} p_y(\lambda) = S$ (4.15), se deducen inmediatamente a partir de (4.44) y (4.42) los resultados (4.45), (4.46), (4.47), (4.48). \square

La figura 4.8 muestra el comportamiento de $C^{(m)}(\lambda)$ y $D^{(m)}(\lambda)$ que acabamos de probar.

A continuación pasamos a enunciar formalmente y demostrar el resultado que hemos avanzado anteriormente acerca del comportamiento de la función objetivo f .

Teorema 4.2 (Comportamiento de la función objetivo como función de λ). *Fijados el resto de parámetros, la función objetivo del problema del precontrol óptimo (4.5) f es continua y diferenciable como función de λ en todo el dominio de λ , $[2, +\infty[$. Además, f es unimodal, en el siguiente sentido:*

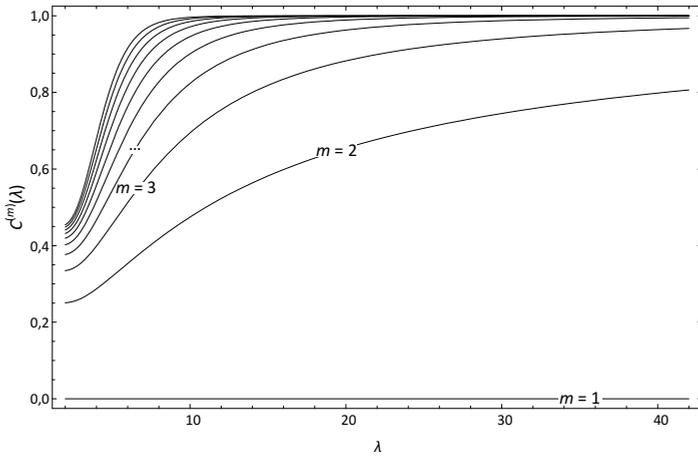
$$\lambda_1, \lambda_2 \in [2, +\infty[: f'(\lambda_1) < 0, \lambda_2 > \lambda_1 \quad \Rightarrow \quad f'(\lambda_2) < 0; \quad (4.49)$$

y se verifica

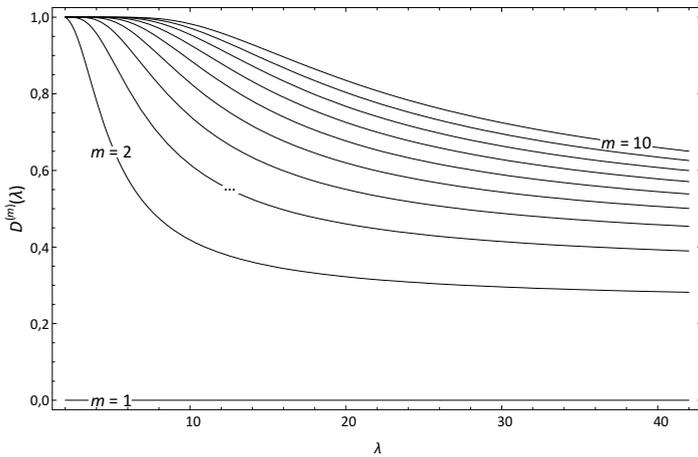
$$f(\lambda = 2) = \frac{1 - S^k}{1 - S}, \quad (4.50)$$

$$\lim_{\lambda \rightarrow +\infty} f(\lambda) = \frac{1 - S^t}{1 - S}. \quad (4.51)$$

Todo ello se cumple también para $f(\cdot | \delta) = E[n(\delta)]$, número esperado de piezas a muestrear cuando el proceso presenta una desviación estandarizada δ , cualquiera que sea el valor del parámetro δ .



(a)



(b)

Figura 4.8: Comportamiento de las funciones auxiliares $C^{(m)}(\lambda)$ y $D^{(m)}(\lambda)$ para diferentes valores de m , en un proceso con desviación $\delta = 0$ y capacidad potencial $C_p = 1$.

Demostración. Haremos uso principalmente de la expresión (3.24) para $f(\lambda)$:

$$f(\lambda) = \frac{\sum_{i=0}^{k-1} p_g(\lambda)^i \sum_{j=0}^{t-1} p_y(\lambda)^j}{1 - \sum_{i=1}^{k-1} p_g(\lambda)^i \sum_{j=1}^{t-1} p_y(\lambda)^j},$$

y también de los resultados (3.20) y (2.14) acerca de la suma parcial de una serie geométrica de razón $r \neq 1$:

$$\sum_{i=0}^{m-1} r^i = \frac{1 - r^m}{1 - r}, \quad \sum_{i=1}^{m-1} r^i = \sum_{i=0}^{m-1} r^i - 1.$$

Retomamos la expresión $Y(\lambda)$ (4.30) utilizada en la demostración del teorema 4.1 y definimos una análoga $G(\lambda)$ para referirnos de manera abreviada a las siguientes sumas parciales:

$$G(\lambda) = \sum_{i=0}^{k-1} p_g(\lambda)^i \tag{4.52}$$

por (3.20):
$$= \frac{1 - p_g(\lambda)^k}{1 - p_g(\lambda)}, \tag{4.53}$$

$$Y(\lambda) = \sum_{j=0}^{t-1} p_y(\lambda)^j$$

por (3.20):
$$= \frac{1 - p_y(\lambda)^t}{1 - p_y(\lambda)}. \tag{4.54}$$

A partir de (3.24) y del hecho de que p_g y p_y son continuas y diferenciables como funciones de λ (véanse la proposición 4.1 y el corolario 4.1), se deduce de manera inmediata que f es también continua y diferenciable. El único posible punto de discontinuidad en f se produciría donde se anulase su denominador, cosa que ha sido ya descartada en la demostración del teorema 4.1 (pues los denominadores y los dominios de

$f(\lambda)$ y $p_Q(\lambda)$ coinciden)⁷.

Basándonos en los mismos resultados, podemos deducir también el comportamiento de f en los extremos del dominio de λ :

$$\begin{aligned} \text{por (3.24):} \quad f(2) &= \frac{\sum_{i=0}^{k-1} p_g(2)^i \sum_{j=0}^{t-1} p_y(2)^j}{1 - \sum_{i=1}^{k-1} p_g(2)^i \sum_{j=1}^{t-1} p_y(2)^j} \\ \text{por (4.14) y (2.14):} \quad &= \sum_{i=0}^{k-1} p_g(2)^i \\ \text{por (4.9) y (3.20):} \quad &= \frac{1 - S^k}{1 - S}, \end{aligned}$$

y de manera análoga, usando (3.24), (4.10) y (4.15), obtenemos:

$$\lim_{\lambda \rightarrow +\infty} f(\lambda) = \sum_{j=0}^{t-1} \lim_{\lambda \rightarrow +\infty} p_y(\lambda)^j = \frac{1 - S^t}{1 - S},$$

con lo que quedan probadas las igualdades (4.50) y (4.51).

Abordamos ahora la demostración del comportamiento *unimodal* de $f(\lambda)$ en todo su dominio $[2, +\infty[$. En primer lugar obtendremos una expresión para la derivada de f respecto de λ , para después probar que si $f'(\lambda_1) < 0$ y $\lambda_2 > \lambda_1$, entonces también $f'(\lambda_2) < 0$, con $\lambda_1, \lambda_2 \in [2, +\infty[$.

Expresamos de manera más compacta la función $f(\lambda)$ (3.24) utilizando las expresiones $G(\lambda)$ (4.52) e $Y(\lambda)$ (4.30) y la igualdad (2.14):

$$\begin{aligned} f(\lambda) &= \frac{G(\lambda)Y(\lambda)}{1 - (G(\lambda) - 1)(Y(\lambda) - 1)} \\ &= \frac{G(\lambda)Y(\lambda)}{G(\lambda) + Y(\lambda) - G(\lambda)Y(\lambda)} \end{aligned}$$

⁷Además, como ya se ha hecho notar en aquel caso, la discontinuidad por la indeterminación de tipo 0^0 que se produce en el término $\sum_{j=0}^{t-1} p_y(\lambda)^j$ cuando $\lambda = 2$ es evitable, pudiendo asumirse que dicho término toma el valor 1.

$$\begin{aligned}
 &= \frac{1}{\frac{1}{G(\lambda)} + \frac{1}{Y(\lambda)} - 1} \\
 &= \left(G(\lambda)^{-1} + Y(\lambda)^{-1} - 1 \right)^{-1}. \tag{4.55}
 \end{aligned}$$

Derivamos $f(\lambda)$ a partir de la expresión (4.55):

$$\begin{aligned}
 f'(\lambda) &= - \left(G(\lambda)^{-1} + Y(\lambda)^{-1} - 1 \right)^{-2} \left(-G'(\lambda)G(\lambda)^{-2} - Y'(\lambda)Y(\lambda)^{-2} \right) \\
 &= \frac{\left(\frac{G'(\lambda)}{G(\lambda)^2} + \frac{Y'(\lambda)}{Y(\lambda)^2} \right)}{\left(\frac{1}{G(\lambda)} + \frac{1}{Y(\lambda)} - 1 \right)^2}. \tag{4.56}
 \end{aligned}$$

Calculamos la derivada de $G(\lambda)$ a partir de su expresión simplificada (4.53):

$$\begin{aligned}
 G'(\lambda) &= \left(\frac{1 - p_g(\lambda)^k}{1 - p_g(\lambda)} \right)' \\
 &= \frac{-kp_g(\lambda)^{k-1} p_g'(\lambda) \cdot (1 - p_g(\lambda)) + (1 - p_g(\lambda)^k) p_g'(\lambda)}{(1 - p_g(\lambda))^2} \\
 &= p_g'(\lambda) \frac{1 - kp_g(\lambda)^{k-1} + (k-1)p_g(\lambda)^k}{(1 - p_g(\lambda))^2}.
 \end{aligned}$$

Por tanto:

$$\begin{aligned}
 \frac{G'(\lambda)}{G(\lambda)^2} &= p_g'(\lambda) \frac{\frac{1 - kp_g(\lambda)^{k-1} + (k-1)p_g(\lambda)^k}{(1 - p_g(\lambda))^2}}{\left(\frac{1 - p_g(\lambda)^k}{1 - p_g(\lambda)} \right)^2} \\
 &= p_g'(\lambda) \frac{1 - kp_g(\lambda)^{k-1} + (k-1)p_g(\lambda)^k}{(1 - p_g(\lambda)^k)^2}. \tag{4.57}
 \end{aligned}$$

A partir de (4.57) y teniendo en cuenta la definición de las funciones

$R^{(m)}$ (4.42) y $C^{(m)}$ (4.44) vistas respectivamente en los lemas 4.4 y 4.5, el cociente $G'(\lambda)/G(\lambda)^2$ puede expresarse de la siguiente forma:

$$\frac{G'(\lambda)}{G(\lambda)^2} = p_g'(\lambda)R^{(k)}(p_g(\lambda)) = p_g'(\lambda)C^{(k)}(\lambda). \quad (4.58)$$

De manera análoga, y teniendo en cuenta la definición en el lema 4.5 de la función $D^{(m)}$ (4.44) y que $p_y'(\lambda) = -p_g'(\lambda)$ (4.16), tenemos:

$$\frac{Y'(\lambda)}{Y(\lambda)^2} = -p_g'(\lambda)R^{(t)}(p_y(\lambda)) = -p_g'(\lambda)D^{(t)}(\lambda). \quad (4.59)$$

Uniendo (4.56), (4.58), (4.59) y (4.44) se llega a la siguiente expresión para $f'(\lambda)$:

$$f'(\lambda) = \frac{p_g'(\lambda)}{\left(\frac{1}{G(\lambda)} + \frac{1}{Y(\lambda)} - 1\right)^2} \left(C^{(k)}(\lambda) - D^{(t)}(\lambda)\right). \quad (4.60)$$

Por tanto, y dado que, según la proposición 4.1, $p_g'(\lambda) < 0$ para todo $\lambda \in [2, +\infty[$, a partir de (4.60) tenemos:

$$f'(\lambda) < 0 \Leftrightarrow C^{(k)}(\lambda) - D^{(t)}(\lambda) > 0 \Leftrightarrow C^{(k)}(\lambda) > D^{(t)}(\lambda). \quad (4.61)$$

Supongamos ahora que existe un determinado valor $\lambda_1 \in [2, +\infty[$ tal que $f'(\lambda_1) < 0$. Por (4.61), eso es si y sólo si

$$C^{(k)}(\lambda_1) > D^{(t)}(\lambda_1). \quad (4.62)$$

Ahora, tomemos un valor cualquiera $\lambda_2 \in [2, +\infty[$ tal que $\lambda_2 > \lambda_1$. Queremos demostrar que también $f'(\lambda_2) < 0$ (es decir, que una vez que la pendiente de $f(\lambda)$ se vuelve negativa, el valor de la función ya no vuelve a aumentar, sino que continúa disminuyendo). Veámoslo.

Efectivamente, en el lema 4.5 se ha demostrado que $C^{(m)}$ y $D^{(m)}$ son funciones continuas y, de modo respectivo, creciente y decreciente

(estrictamente si $m \geq 2$). Usando el lema 4.5, $\lambda_1 < \lambda_2$ y (4.62), tenemos:

$$C^{(k)}(\lambda_2) \geq C^{(k)}(\lambda_1) > D^{(t)}(\lambda_1) \geq D^{(t)}(\lambda_2);$$

es decir, partiendo de que $f'(\lambda_1) < 0$ y tomando $\lambda_2 > \lambda_1$ deducimos que $C^{(k)}(\lambda_2) > D^{(t)}(\lambda_2)$. Según (4.61), eso sólo puede ser si $f'(\lambda_2) < 0$, como se quería demostrar.

Nótese que toda la demostración es igualmente válida para el caso en que en la definición de f (3.25) el parámetro δ no estuviera fijado al valor 0, ya que en ningún momento se ha utilizado este hecho. Es decir, el valor esperado de la cantidad de piezas que es necesario muestrear para llevar a cabo el procedimiento del precontrol $E[n(\delta)]$ se comporta tal y como se ha descrito y demostrado en este teorema, independientemente de cuál sea la desviación estandarizada δ del proceso. \square

El resultado que acabamos de demostrar, y que se ilustra en la figura 4.9, caracteriza el comportamiento de la función objetivo f en

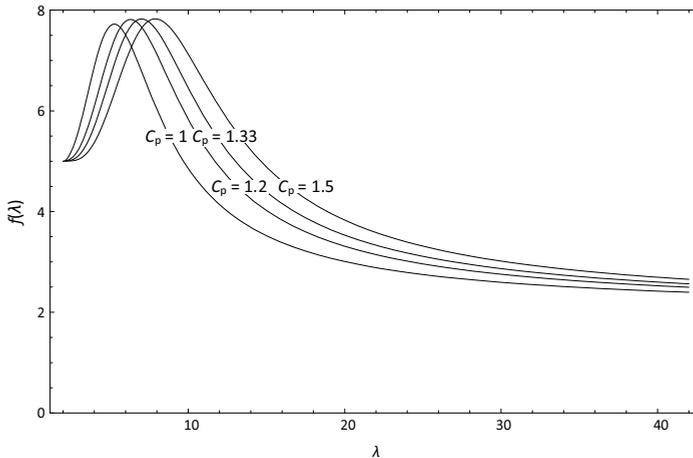


Figura 4.9: Comportamiento de f como función del parámetro λ , en el precontrol con $k = 5$ y $t = 2$, para varios valores del parámetro C_p .

función del parámetro λ . En primer término, sabemos por (2.24) que según aumenta λ se reduce la anchura de la zona verde W_G , y disminuye por tanto la probabilidad de que una pieza sea verde, mientras que aumenta la de que una pieza sea amarilla (recuérdense la proposición 4.1 y el corolario 4.1). Como consecuencia, el efecto que la progresiva reducción de la zona verde tiene en el número esperado de piezas a muestrear hasta finalizar el procedimiento de precontrol (que es lo representado por la función objetivo f) es justamente el que se explica en el teorema anterior: inicialmente, dicha reducción de W_G , la cual hace que se incremente la probabilidad de obtener una pieza amarilla p_y , provoca que la intercalación de verdes y amarillas sea más frecuente, haciendo más difícil por tanto la obtención de k verdes o t amarillas consecutivas (y por eso se produce un incremento de f); hasta que llega un momento en que la probabilidad de verde p_g se hace tan pequeña que la aparición de t amarillas consecutivas comienza a hacerse cada vez más probable (y por eso, una vez disminuye el valor de f éste ya no vuelve a incrementarse, ya que según va aumentando λ y reduciéndose W_G , la probabilidad p_g sigue tomando valores cada vez más cercanos a 0).

Según se puede deducir de esta explicación, la función $f(\lambda)$ alcanzaría su máximo en un único punto $\lambda_{\text{máx}}$ (lo cual se corresponde efectivamente con el concepto de *unimodalidad* más extendido); lo formalizamos mediante el siguiente corolario.

Corolario 4.4 (Caracterización del máximo de la función objetivo como función de λ). *Considerada la función objetivo del problema del precontrol (4.5) como función de λ , fijados los parámetros k y t a sendos valores mayores que 1 y fijados arbitrariamente el resto de parámetros, el máximo de $f(\lambda)$ se alcanza en un único valor $\lambda_{\text{máx}} \in [2, +\infty[$ tal que verifica la igualdad siguiente:*

$$C^{(k)}(\lambda_{\text{máx}}) = D^{(t)}(\lambda_{\text{máx}}), \quad (4.63)$$

siendo $C^{(k)}$ y $D^{(t)}$ las funciones definidas según (4.44) en el lema 4.5.

En consecuencia, el valor $\lambda_{\text{máx}}$ es un máximo (global) estricto, en el sentido

de que

$$f(\lambda_{\text{máx}}) > f(\lambda), \quad \forall \lambda \in [2, +\infty[\setminus \{\lambda_{\text{máx}}\}. \quad (4.64)$$

Asimismo, se cumple

$$f'(\lambda) > 0 \Leftrightarrow C^{(k)}(\lambda) < D^{(t)}(\lambda). \quad (4.65)$$

Por último, se verifica

$$\lambda_1 < \lambda_{\text{máx}} \Leftrightarrow C^{(k)}(\lambda_1) < D^{(t)}(\lambda_1) \quad y \quad (4.66)$$

$$\lambda_{\text{máx}} < \lambda_2 \Leftrightarrow C^{(k)}(\lambda_2) > D^{(t)}(\lambda_2). \quad (4.67)$$

En los casos en que k o t toman el valor 1, las funciones $f(\lambda \mid k = 1, t > 1)$ y $f(\lambda \mid k > 1, t = 1)$ son, de modo respectivo, estrictamente creciente y estrictamente decreciente en $[2, +\infty[$, mientras que $f(\lambda \mid k = 1, t = 1)$ es una función constante.

Demostración. Si $k = 1$, entonces la expresión de f (3.25) queda así:

$$f(\lambda \mid k = 1) = \frac{1 - p_y(\lambda)^t}{1 - p_y(\lambda)} = \sum_{j=0}^{t-1} p_y(\lambda)^j. \quad (4.68)$$

Según el corolario 4.1, $p_y(\lambda)$ es estrictamente creciente; esto implica que la expresión (4.68) es también estrictamente creciente con respecto a λ en el caso en que el parámetro t toma un valor superior a 1, y es constante igual a 1 cuando $t = 1$.

Análogamente, si $t = 1$, a partir de (3.25) se tiene

$$f(\lambda \mid t = 1) = \frac{1 - p_g(\lambda)^k}{1 - p_g(\lambda)} = \sum_{i=0}^{k-1} p_g(\lambda)^i. \quad (4.69)$$

Dado que p_g es estrictamente decreciente como función de λ (proposición 4.1), la expresión (4.69) necesariamente también lo es, siempre que $k > 1$. Así, queda probado el comportamiento de f en estos casos particulares.

Retomamos la expresión (4.60) para la derivada de $f(\lambda)$ que se ha

obtenido en la demostración del teorema 4.1:

$$f'(\lambda) = \frac{p_g'(\lambda)}{\left(\frac{1}{G(\lambda)} + \frac{1}{Y(\lambda)} - 1\right)^2} \left(C^{(k)}(\lambda) - D^{(t)}(\lambda)\right),$$

donde las definiciones de $C^{(k)}$ y $D^{(t)}$ (4.44) se han recordado en el enunciado de este corolario, y las funciones G (4.52) e Y (4.30) representan las siguientes sumas parciales:

$$G(\lambda) = \sum_{i=0}^{k-1} p_g(\lambda)^i = \frac{1 - p_g(\lambda)^k}{1 - p_g(\lambda)},$$

$$Y(\lambda) = \sum_{j=0}^{t-1} p_y(\lambda)^j = \frac{1 - p_y(\lambda)^t}{1 - p_y(\lambda)}.$$

A partir de la expresión (4.60), y teniendo en cuenta que, de acuerdo con la proposición 4.1, $p_g'(\lambda) < 0$ para todo $\lambda \in [2, +\infty[$, se deducen de manera inmediata los siguientes resultados. En primer lugar,

$$f'(\lambda) < 0 \Leftrightarrow C^{(k)}(\lambda) > D^{(t)}(\lambda),$$

lo cual ya se utilizó en la demostración del teorema 4.1 (véase (4.61)); también

$$f'(\lambda) > 0 \Leftrightarrow C^{(k)}(\lambda) < D^{(t)}(\lambda),$$

con lo que queda demostrada la doble implicación (4.65); y por último

$$f'(\lambda) = 0 \Leftrightarrow C^{(k)}(\lambda) = D^{(t)}(\lambda). \quad (4.70)$$

Nótese que el desarrollo de $f'(\lambda)$ (4.60) se ha obtenido en la demostración del teorema 4.1 sin realizar ninguna suposición sobre los valores de k y t ; por tanto, todo lo que acabamos de deducir es también válido para los casos particulares en que k o t toman el valor 1.

Supongamos ahora que $k, t \geq 2$.

A continuación probaremos que hay un único valor λ_0 tal que en él

$C^{(k)}$ y $D^{(t)}$ toman el mismo valor.

Consideremos la función auxiliar

$$v(\lambda) = C^{(k)}(\lambda) - D^{(t)}(\lambda), \quad (4.71)$$

definida en $[2, +\infty[$. A partir de las propiedades de $C^{(k)}$ y $D^{(t)}$ (lema 4.5) puede deducirse que v es continua en todo su dominio y que además cumple

$$\begin{aligned} v(2) &= C^{(k)}(2) - D^{(t)}(2) = R^{(k)}(S) - 1 < 0 \\ \lim_{\lambda \rightarrow +\infty} v(\lambda) &= \lim_{\lambda \rightarrow +\infty} C^{(k)}(\lambda) - \lim_{\lambda \rightarrow +\infty} D^{(t)}(\lambda) = 1 - R^{(t)}(S) > 0 \end{aligned}$$

donde hemos aplicado que $R^{(m)}$ es una función continua y decreciente si $m > 1$ (lema 4.4), que $R^{(m)}(0) = 1$ para todo m y que $S > 0$ (4.4).

La función v comienza, pues, tomando un valor negativo en $\lambda = 2$, y en algún momento pasa a tomar valores positivos. Por el teorema de Bolzano, sabemos que necesariamente en algún momento la gráfica de v cruzará el eje de abscisas, es decir, *existe* un valor $\lambda_0 \in]2, +\infty[$ que verifica $v(\lambda_0) = 0$, o lo que es lo mismo según (4.71),

$$C^{(k)}(\lambda_0) = D^{(t)}(\lambda_0). \quad (4.72)$$

Veamos ahora que no hay más que un punto en $[2, +\infty[$ que verifique la propiedad (4.72); es consecuencia del comportamiento respectivo estrictamente creciente y decreciente de $C^{(k)}$ y $D^{(t)}$.

Por reducción al absurdo, supongamos que existe un valor $\lambda_1 \in [2, +\infty[$, $\lambda_1 \neq \lambda_0$, tal que

$$C^{(k)}(\lambda_1) = D^{(t)}(\lambda_1). \quad (4.73)$$

En concreto, supongamos también, sin pérdida de generalidad, que $\lambda_0 < \lambda_1$.

Entonces, por ser $C^{(k)}$ y $D^{(t)}$ continuas y, de modo respectivo, estrictamente creciente y estrictamente decreciente como función de λ (lema 4.5),

y considerando cierto (4.73), tenemos:

$$C^{(k)}(\lambda_0) < C^{(k)}(\lambda_1) = D^{(t)}(\lambda_1) < D^{(t)}(\lambda_0).$$

Es decir, del hecho de suponer cierto (4.73) llegamos a la conclusión de que $C^{(k)}(\lambda_0) \neq D^{(t)}(\lambda_0)$, lo cual entra en contradicción con (4.72). Por reducción al absurdo, concluimos que no puede existir tal valor λ_1 ; y por tanto, λ_0 es el *único* valor que verifica la igualdad (4.72).

Según (4.70), lo que acabamos de probar es que existe un único punto λ_0 tal que $f'(\lambda_0) = 0$. A continuación, comprobaremos que dicho punto estacionario es efectivamente un máximo.

Dado cualquier valor $\lambda_1 < \lambda_0$, por ser $C^{(k)}$ estrictamente creciente y $D^{(t)}$ estrictamente decreciente y ser ambas continuas (lema 4.5), y teniendo en cuenta (4.72):

$$C^{(k)}(\lambda_1) < C^{(k)}(\lambda_0) = D^{(t)}(\lambda_0) < D^{(t)}(\lambda_1),$$

esto es, $C^{(k)}(\lambda_1) < D^{(t)}(\lambda_1)$. Según la equivalencia (4.65) que ya hemos probado, esto es así si y sólo si $f'(\lambda_1) > 0$. Y esto es verdad para cualquier λ_1 tal que $\lambda_1 < \lambda_0$, $\lambda_1 \in [2, +\infty[$.

Análogamente, para cualquier valor $\lambda_2 > \lambda_0$, utilizando de nuevo las propiedades de $C^{(k)}$ y $D^{(t)}$ y la igualdad (4.72) se tiene:

$$C^{(k)}(\lambda_2) > C^{(k)}(\lambda_0) = D^{(t)}(\lambda_0) > D^{(t)}(\lambda_2),$$

es decir, $C^{(k)}(\lambda_2) > D^{(t)}(\lambda_2)$, o lo que es lo mismo según (4.61), $f'(\lambda_2) < 0$.

En resumen, λ_0 es un punto estacionario de f tal que

$$f'(\lambda_1) > 0 \quad \forall \lambda_1 < \lambda_0, \lambda_1 \in [2, +\infty[\quad \text{y} \quad (4.74)$$

$$f'(\lambda_2) < 0 \quad \forall \lambda_2 > \lambda_0, \lambda_2 \in [2, +\infty[. \quad (4.75)$$

En consecuencia, $\lambda_0 = \lambda_{\text{máx}}$ es un *máximo* de f ; y además es *único*, según se ha demostrado ya más arriba.

Por ser un máximo de f , $\lambda_{\text{máx}}$ cumple $f(\lambda_{\text{máx}}) \geq f(\lambda)$ para todo λ en

un cierto entorno de $\lambda_{\text{máx}}$. Pero de la unicidad de $\lambda_{\text{máx}}$ y de (4.74) y (4.75) se deduce directamente que no puede haber ningún valor $\lambda \neq \lambda_{\text{máx}}$ en $]2, +\infty[$ tal que $f(\lambda) = f(\lambda_{\text{máx}})$. Esto demuestra que es cierta la expresión (4.64), es decir, que $\lambda_{\text{máx}}$ es un máximo (global) en sentido estricto.

Únicamente nos resta probar las implicaciones de derecha a izquierda de las dobles implicaciones (4.66) y (4.67). Es inmediato.

Dado $\lambda_1 \in]2, +\infty[$ tal que $C^{(k)}(\lambda_1) < D^{(t)}(\lambda_1)$, entonces $f'(\lambda_1) > 0$ por (4.65). Según se deduce de (4.75), λ_1 no puede ser mayor que λ_0 ya que entonces la derivada de f en λ_1 tendría signo negativo. Tampoco es posible $\lambda_1 = \lambda_0$, ya que λ_0 cumple por definición $C^{(k)}(\lambda_0) = D^{(t)}(\lambda_0)$. Por tanto, necesariamente $\lambda_1 < \lambda_0 = \lambda_{\text{máx}}$.

Análogamente se demuestra que si $C^{(k)}(\lambda_2) > D^{(t)}(\lambda_2)$, entonces $\lambda_2 > \lambda_0 = \lambda_{\text{máx}}$. \square

La figura 4.10 ilustra la relación que terminamos de probar.

Incluimos un último resultado, consecuencia directa de todo lo anterior, en el que se explicita lo que desde un primer momento estábamos interesados en probar: que no existe ningún *valle* en la gráfica de $f(\lambda)$.

Corolario 4.5. *Considerada la función objetivo del problema del precontrol (4.5) como función de λ y fijados arbitrariamente el resto de parámetros, exceptuando el caso en que $k = t = 1$, no existe ningún valor $\lambda_{\text{mín}} \in]2, +\infty[$ que sea un mínimo (local) de f , esto es, que verifique*

$$\exists \epsilon > 0 \text{ tal que } f(\lambda_{\text{mín}}) \leq f(\lambda) \quad \forall \lambda \in]\lambda_{\text{mín}} - \epsilon, \lambda_{\text{mín}} + \epsilon[. \quad (4.76)$$

Demostración. Por reducción al absurdo, supongamos que existe un valor $\lambda_{\text{mín}} \in]2, +\infty[$ que verifica (4.76).

Dado que la función $f(\lambda)$ es derivable en todo su dominio $]2, +\infty[$ (teorema 4.2) y que $\lambda_{\text{mín}}$ se encuentra en el interior de dicho dominio, por el teorema de Fermat sobre puntos estacionarios sabemos que si $\lambda_{\text{mín}}$ cumple (4.76), entonces

$$f'(\lambda_{\text{mín}}) = 0. \quad (4.77)$$

En el corolario anterior se ha demostrado que $f(\lambda)$ es estrictamente

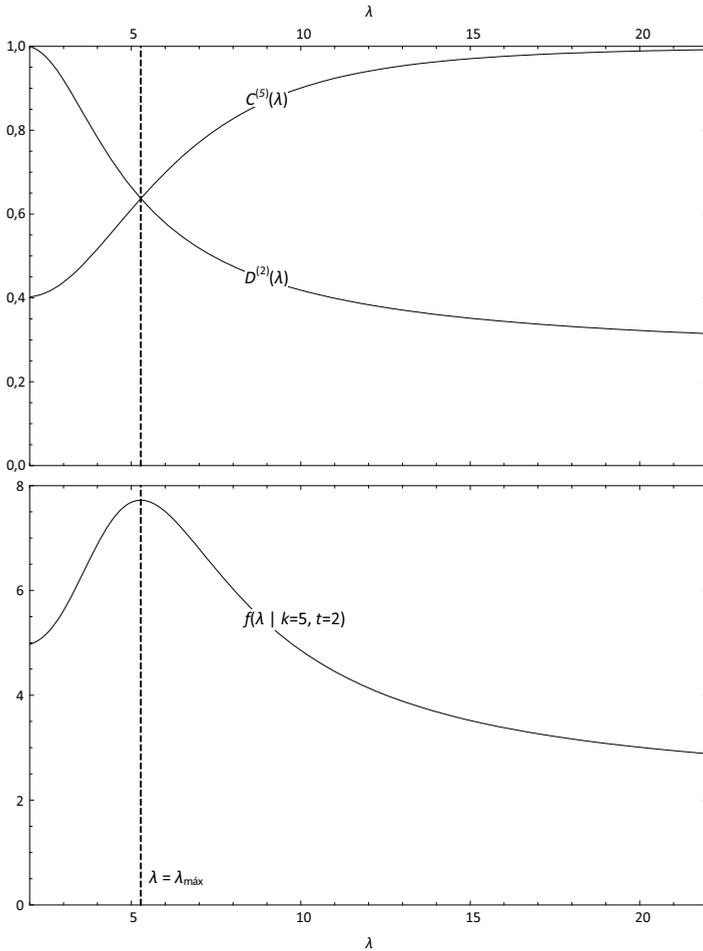


Figura 4.10: Caracterización del máximo de $f(\lambda)$ como el punto en que $C^{(k)}(\lambda) = D^{(t)}(\lambda)$, en el precontrol con $k = 5$ y $t = 2$, para un proceso con capacidad potencial $C_p = 1$.

creciente cuando $k = 1, t > 1$, y estrictamente decreciente si $t = 1, k > 1$. Por tanto, en ambos casos es imposible que exista un valor $\lambda_{\text{mín}}$ que verifique (4.77).

En el caso en que los parámetros k y t estén fijados a sendos valores mayores o iguales a 2, en la demostración del corolario anterior se ha

visto que el único punto $\lambda_0 \in [2, +\infty[$ tal que $f'(\lambda_0) = 0$ es $\lambda_0 = \lambda_{\text{máx}}$.

Por tanto, suponiendo que es cierto $f'(\lambda_{\text{mín}}) = 0$, llegamos a la conclusión de que $\lambda_{\text{mín}} = \lambda_{\text{máx}}$. Pero esto nos conduce a una contradicción, ya que, por un lado, según (4.76), $f(\lambda_{\text{mín}}) \leq f(\lambda)$ para todo λ en un cierto entorno de $\lambda_{\text{mín}}$, mientras que, por el corolario 4.4, $\lambda_{\text{máx}}$ es un máximo en sentido estricto de f , y por tanto cumple todo lo contrario: $f(\lambda_{\text{máx}}) > f(\lambda)$ para todo λ en un cierto entorno de $\lambda_{\text{máx}}$ (en realidad, en todo $[2, +\infty[\setminus \{\lambda_{\text{máx}}\}$, según (4.64)).

En todos los casos considerados, la suposición de que existe un valor $\lambda_{\text{mín}}$ tal como el que se describe en el enunciado conduce a una contradicción. Por reducción al absurdo, concluimos que no existe un valor $\lambda_{\text{mín}}$ con tales propiedades, como se quería demostrar.

La demostración de este resultado también se podría haber abordado utilizando directamente el teorema 4.2: una vez $f(\lambda)$ toma una pendiente negativa, continúa decreciendo indefinidamente, lo cual es incompatible con la existencia de un mínimo. \square

Únicamente en el caso trivial en que $k = t = 1$, la función objetivo f es constante igual a 1 como función de λ , ya que siempre se necesitará exactamente *una* pieza (verde, amarilla o roja) para finalizar el proceso del precontrol. En el resto de casos, acabamos de demostrar finalmente que el número esperado de piezas a muestrear como función del parámetro λ no presenta un mínimo en $]2, +\infty[$. Este resultado será fundamental más adelante, cuando abordemos la optimización de la función objetivo del problema OPC (4.5) fijados los parámetros k y t .

Corolario 4.6 (No convexidad del problema OPC). *Dados $\alpha^* \in]0, 1[$, $\beta^* \in]0, 1[$ y $\delta^* > 0$, y conocido el valor del parámetro $C_p > 0$, el problema de optimización no lineal entera mixta OPC (4.5) es no convexo.*

Demostración. Un problema de MINLP se dice que es *convexo* si y sólo si las funciones que lo componen (función objetivo y lados izquierdos de las restricciones —expresadas de forma que los respectivos lados derechos sean constantes—) son convexas. En caso contrario, se dice que es *no convexo* (véase, por ejemplo, [Burer y Letchford, 2012](#)).

Una función $F : \mathbb{R}^p \rightarrow \mathbb{R}$ es *convexa* si para todo par de puntos $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^p$ y todo $\theta \in [0, 1]$ se cumple

$$F(\theta \mathbf{x}_1 + (1 - \theta) \mathbf{x}_2) \leq \theta F(\mathbf{x}_1) + (1 - \theta)F(\mathbf{x}_2), \quad (4.78)$$

es decir, si el segmento que une dos puntos cualesquiera del grafo de F nunca queda por debajo del propio grafo. Si F no está definida en todo \mathbb{R}^p , entonces debe exigirse también que el dominio de F sea un conjunto convexo (véase a continuación). La definición es válida también para el caso de funciones de \mathbb{R} en \mathbb{R} , simplemente tomando $p = 1$.

Asimismo, un conjunto $C \in \mathbb{R}^p$ se dice que es *convexo* si el segmento que une cualquier par de puntos de C está contenido en el mismo conjunto. Una función es convexa si y sólo si su epigrafo (el conjunto de puntos que quedan por encima de su grafo, incluido éste) es un conjunto convexo. Estos y otros conceptos básicos relacionados con la convexidad de funciones pueden encontrarse, por ejemplo, en [Boyd y Vandenberghe \(2004\)](#).

Probaremos que la función objetivo f de (4.5) no es convexa, como función de varias variables, y que, por tanto, tampoco lo es el problema.

Sean $k_0, t_0 \in \mathbb{Z}$, $k_0, t_0 > 1$. Fijados $k = k_0$ y $t = t_0$, consideramos la función objetivo del problema (4.5) como función de λ , $f(\lambda | k_0, t_0, C_p)$.

Una función real de variable real diferenciable, con derivada continua y presencia de máximos estrictos es no convexa ([Hiriart-Urruty y Lemaréchal, 2001](#)). La función $f(\lambda)$ presenta un máximo en sentido estricto, según el corolario 4.4; y es fácil deducir que su derivada es continua a partir de, por ejemplo, la expresión (4.60) obtenida para ella en el teorema 4.2. Gracias a todo ello se tiene, por tanto, que $f(\lambda | k_0, t_0, C_p)$ no es convexa.

Una interesante propiedad ([Boyd y Vandenberghe, 2004](#)) afirma que una función $F : \mathbb{R}^p \rightarrow \mathbb{R}$ es convexa si y sólo si lo es como función real de variable real a lo largo de cualquier segmento contenido en su dominio, es decir, si y sólo si para todo $\mathbf{x} \in \mathbb{R}^p$ perteneciente al dominio de F y para toda *dirección* $\mathbf{v} \in \mathbb{R}^p$, la función $G(\theta) = F(\mathbf{x} + \theta \mathbf{v})$ es convexa.

Por lo tanto, basta con tomar $\mathbf{x} = (k_0, t_0, 0)$ y $\mathbf{v} = (0, 0, 1)$. Efectivamente, hemos deducido arriba que $f(\lambda | k_0, t_0, C_p) = f(\mathbf{x} + \lambda \mathbf{v} | C_p)$, $\lambda \geq 2$, es *no convexa* y, por la propiedad enunciada, tampoco lo es f como función de varias variables, que es lo que se quería demostrar. □

4.3.2. Comportamiento como función de k

A continuación abordamos el estudio del comportamiento de las expresiones involucradas en el modelo del precontrol como funciones de la variable k , fijados el resto de parámetros.

De acuerdo con la notación introducida en la subsección 4.1.1, al estar considerando el parámetro del precontrol k como variable y como fijos los valores de t , λ , C_p y δ , expresaremos la probabilidad de cualificar el proceso en función de k como $p_Q(k)$, es decir,

$$p_Q(k) = p_Q(k | t, \lambda, C_p, \delta);$$

y análogamente para las expresiones $\alpha(k)$, $\beta(k)$ y $f(k)$ que aparecerán más adelante. Nótese, además, que las probabilidades de obtener una pieza verde, amarilla y roja son independientes del valor del parámetro k , por lo que serán nombradas como p_g , p_y y p_r , respectivamente.

Proposición 4.2 (Comportamiento de la probabilidad de cualificar como función de k). *Fijados el resto de parámetros, la probabilidad de cualificar p_Q disminuye según aumenta el valor de k , en todo el dominio de k ; es decir:*

$$p_Q(k+1) < p_Q(k), \quad \text{para todo } k \in \mathbb{Z}, k \geq 1. \quad (4.79)$$

Concretamente, se cumple:

$$p_Q(k+1) = p_Q(k) \left(p_g + \frac{p_g^{k+1} \sum_{j=1}^{t-1} p_y^j}{1 - \sum_{j=1}^{t-1} p_y^j \sum_{i=1}^k p_g^i} \right), \quad \text{para todo } k \in \mathbb{Z}, k \geq 1. \quad (4.80)$$

Además, se verifica

$$p_Q(k = 1) = p_g \sum_{j=0}^{t-1} p_y^j, \quad (4.81)$$

$$\lim_{k \rightarrow +\infty} p_Q(k) = 0. \quad (4.82)$$

Demostración. Basaremos la demostración en la idea de que para obtener $k + 1$ piezas verdes consecutivas es *necesario* alcanzar antes el *hito* de tener k verdes consecutivas.

Conviene remarcar que, considerando un plan de precontrol determinado (k, t, λ) , el evento *cualificar* (cuya probabilidad se expresa por p_Q) representa el hecho de *observar k verdes consecutivas en una extracción sucesiva de piezas sin antes observar t amarillas consecutivas ni 1 roja*.

Retomamos la notación empleada en la subsección 3.3.2, según la cual el evento iG se produce cuando la última pieza extraída ha sido verde y con ella se acumulan i extracciones consecutivas de piezas verdes, y el evento jY se produce cuando la última pieza extraída ha sido amarilla y con ella se acumulan j extracciones consecutivas de piezas amarillas.

A partir de todo esto, es claro que podemos relacionar $p_Q(k + 1)$ y $p_Q(k)$ de la siguiente forma:

$$p_Q(k + 1) = p_Q(k)p_Q(k + 1 | kG). \quad (4.83)$$

La probabilidad $p_Q(k + 1 | kG)$ es estrictamente menor que 1, ya que, entre otros motivos, la probabilidad de obtener una pieza roja nunca es nula según (3.5).

Usando $p_Q(k + 1 | kG) < 1$ y (4.83) concluimos:

$$p_Q(k + 1) < p_Q(k),$$

para cualquier $k \in \mathbb{Z}, k \geq 1$, como se quería demostrar.

A partir del trabajo desarrollado por [San Matías et al. \(2004\)](#) es inmediato deducir una expresión explícita para $p_Q(k + 1 | kG)$. En efecto,

según [San Matías et al.](#) (adaptando la notación):

$$p_Q(k+1 | iG) = p_g p_Q(k+1 | (i+1)G) + p_y p_Q(k+1 | 1Y), \quad \forall i \leq k, \quad (4.84)$$

$$p_Q(k+1 | (k+1)G) = 1, \quad (4.85)$$

$$p_Q(k+1 | 1Y) = p_g p_Q(k+1 | 1G) \sum_{j=0}^{t-2} p_y^j, \quad (4.86)$$

$$p_Q(k+1 | 1G) = \frac{p_g^k}{1 - \sum_{j=1}^{t-1} p_y^j \sum_{i=1}^k p_g^i}. \quad (4.87)$$

Enlazando (4.84), (4.85), (4.86) y (4.87) se tiene:

$$p_Q(k+1 | kG) = p_g + p_y \frac{p_g^{k+1} \sum_{j=0}^{t-2} p_y^j}{1 - \sum_{j=1}^{t-1} p_y^j \sum_{i=1}^k p_g^i},$$

con lo que, por (4.83), queda también probada la igualdad (4.80).

La demostración de las igualdades (4.81) y (4.82) es inmediata a partir de las expresiones (2.13) y (2.12):

$$p_Q(k | k = 1 \text{ o } t = 1) = p_g^k \sum_{j=0}^{t-1} p_y^j$$

$$p_Q(k) = p_g^k \frac{\sum_{j=0}^{t-1} p_y^j}{1 - \sum_{j=1}^{t-1} p_y^j \sum_{i=1}^{k-1} p_g^i},$$

usando (2.15) $\sum_{i=0}^{+\infty} r^i = \frac{1}{1-r}$, $\forall r : |r| < 1$. □

La lógica del resultado anterior es clara: si se incrementa el número de piezas verdes consecutivas exigido para cualificar el proceso, la

probabilidad de cualificar disminuye, tendiendo asintóticamente a cero. A partir de él se derivan de manera inmediata los siguientes resultados:

Corolario 4.7 (Comportamiento de α como función de k). *Fijados el resto de parámetros, la probabilidad de falsa alarma α se incrementa según aumenta el valor de k , en todo el dominio de k ; es decir:*

$$\alpha(k+1) > \alpha(k), \quad \text{para todo } k \in \mathbb{Z}, k \geq 1. \quad (4.88)$$

Además,

$$\alpha(k=1) = 1 - p_g(\delta=0) \sum_{j=0}^{t-1} p_y(\delta=0)^j, \quad (4.89)$$

$$\lim_{k \rightarrow +\infty} \alpha(k) = 1. \quad (4.90)$$

Demostración. El resultado es inmediato usando la proposición 4.2 que acabamos de demostrar, y recordando la definición de α (2.20): $\alpha(k, t, \lambda | C_p) = 1 - p_Q(k, t, \lambda | C_p, \delta=0)$. \square

Corolario 4.8 (Comportamiento de β como función de k). *Fijados el resto de parámetros, la probabilidad β de no detectar una cierta desviación δ del proceso disminuye según aumenta el valor de k , en todo el dominio de k ; es decir:*

$$\beta(k+1 | \delta) < \beta(k | \delta), \quad \text{para todo } k \in \mathbb{Z}, k \geq 1. \quad (4.91)$$

Además,

$$\beta(k=1 | \delta) = p_g(\delta) \sum_{j=0}^{t-1} p_y(\delta)^j, \quad (4.92)$$

$$\lim_{k \rightarrow +\infty} \beta(k | \delta) = 0. \quad (4.93)$$

Demostración. Es también inmediato usando la proposición 4.2 y la definición de β (2.27): $\beta(k, t, \lambda | C_p, \delta) = p_Q(k, t, \lambda | C_p, \delta)$. \square

Las figuras 4.11 y 4.12 ilustran el comportamiento de α y β como

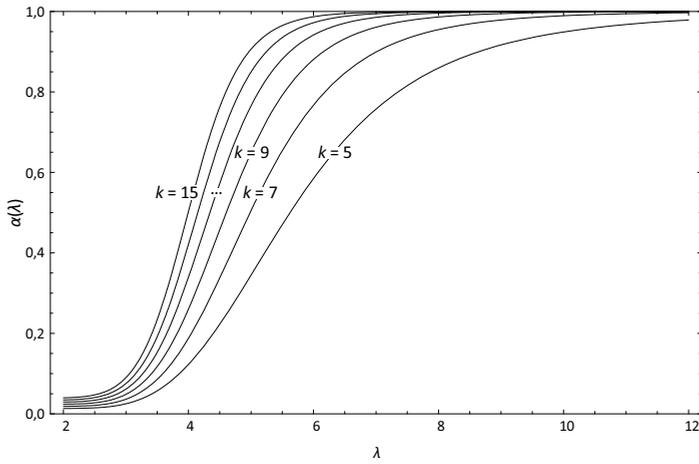


Figura 4.11: Comportamiento de α como función del parámetro k , en el precontrol con $t = 2$, para un proceso con capacidad potencial $C_p = 1$.

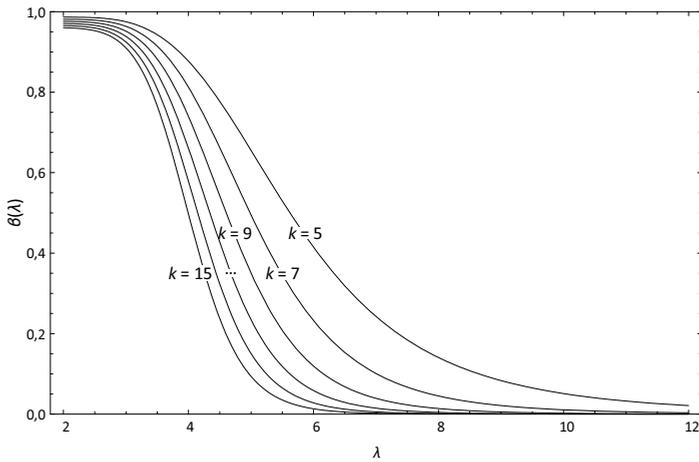


Figura 4.12: Comportamiento de $\beta(\cdot | \delta = 0)$ como función del parámetro k , en el precontrol con $t = 2$, para un proceso con capacidad potencial $C_p = 1$.

función de k al que se refieren los corolarios que terminamos de probar.

Proposición 4.3 (Comportamiento de la función objetivo como función de k). *Fijados el resto de parámetros, el valor de la función objetivo del problema del precontrol óptimo (4.5) f se incrementa según aumenta el valor de k , es decir:*

$$f(k+1) > f(k), \quad \text{para todo } k \in \mathbb{Z}, k \geq 1. \quad (4.94)$$

Además, se cumple:

$$f(k=1) = \sum_{j=0}^{t-1} p_y(\delta=0)^j, \quad (4.95)$$

$$\lim_{k \rightarrow +\infty} f(k) = \frac{1}{\frac{1-p_y(\delta=0)}{1-p_y(\delta=0)^t} - p_g(\delta=0)}. \quad (4.96)$$

Demostración. La igualdad (4.95) ya se probó en la demostración del corolario 4.4 (véase (4.68)).

El resultado (4.96) se deduce de manera inmediata, bien a partir de la expresión para f (3.25) vista en el capítulo 3, o bien a partir de la (4.55) deducida en la demostración del teorema 4.2:

$$f(k) = \frac{1}{\left(\sum_{i=0}^{k-1} p_g(\delta=0)^i\right)^{-1} + \left(\sum_{j=0}^{t-1} p_y(\delta=0)^j\right)^{-1} - 1},$$

usando (2.15) $\sum_{i=0}^{+\infty} r^i = \frac{1}{1-r}$ y (3.20) $\sum_{i=0}^{m-1} r^i = \frac{1-r^m}{1-r}$, $\forall r : |r| < 1$.

La expresión (4.94) se demuestra de modo inmediato usando la expresión para f (4.55) arriba indicada, y teniendo en cuenta que $\sum_{i=0}^{k-1} p_g(\delta=0)^i < \sum_{i=0}^k p_g(\delta=0)^i$ (pues en cualquier caso se cumple $p_g > 0$; véase (2.25)). \square

El resultado anterior, que se ilustra en la figura 4.13 tiene sentido: como se ha visto en la proposición 4.2, según aumenta el valor de k , dejando fijos el resto de parámetros, la probabilidad de cualificar el proceso tiende monótonamente a cero, y en consecuencia el valor de f (número esperado

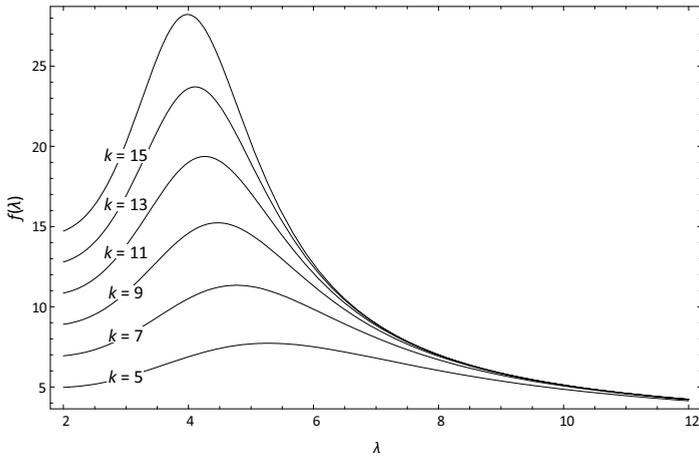


Figura 4.13: Comportamiento de f como función del parámetro k , en el precontrol con $t = 2$, para un proceso con capacidad potencial $C_p = 1$.

de piezas a muestrear para finalizar el procedimiento del precontrol) tiende monótonamente hacia una constante (ya que t lo es): el número de piezas necesario, por término medio, para terminar no cualificando el proceso.

4.3.3. Comportamiento como función de t

A continuación abordamos el estudio del comportamiento de las expresiones involucradas en el modelo del precontrol como funciones de la variable t , fijados el resto de parámetros.

De manera similar a lo hecho en la subsección anterior, denotaremos la probabilidad de cualificar el proceso en función de t como $p_Q(t)$, es decir,

$$p_Q(t) = p_Q(t \mid k, \lambda, C_p, \delta),$$

suponiendo fijados el resto de parámetros; análogamente para las expresiones $\alpha(t)$, $\beta(t)$ y $f(t)$ que aparecerán más adelante. Recuérdese que las probabilidades p_g , p_y y p_r no dependen del valor del parámetro t .

Proposición 4.4 (Comportamiento de la probabilidad de cualificar como función de t). *Fijado el parámetro λ a un valor mayor que 2 y fijados arbitrariamente el resto de parámetros, la probabilidad de cualificar p_Q se incrementa según aumenta el valor de t , en todo el dominio de t ; es decir:*

$$p_Q(t+1) > p_Q(t), \quad \text{para todo } t \in \mathbb{Z}, t \geq 1. \quad (4.97)$$

Además, en ese caso se verifica

$$p_Q(t=1) = p_g^k, \quad (4.98)$$

$$\lim_{t \rightarrow +\infty} p_Q(t) = \frac{p_g^k}{1 - p_y \sum_{i=0}^{k-1} p_g^i}. \quad (4.99)$$

En el caso en que λ toma el valor 2, la probabilidad de cualificar se mantiene constante respecto de t , fijados el resto de parámetros; esto es:

$$p_Q(t_1 | \lambda = 2) = p_Q(t_2 | \lambda = 2), \quad \forall t_1, t_2 \in \mathbb{Z}, t_1, t_2 \geq 1. \quad (4.100)$$

Demostración. Consideraremos la expresión (2.12) para la probabilidad de cualificar $p_Q(t)$:

$$p_Q(t) = p_g^k \frac{\sum_{j=0}^{t-1} p_y^j}{1 - \sum_{j=1}^{t-1} p_y^j \sum_{i=1}^{k-1} p_g^i}.$$

Para el caso $\lambda = 2$, según se ha visto en el teorema 4.1 y de acuerdo con (4.24), la expresión (2.12) queda así:

$$p_Q(t | \lambda = 2) = p_g(\lambda = 2)^k = S^k,$$

para todo $t \in \mathbb{Z}, t \geq 1$, y con ello queda demostrada la igualdad (4.100).

Según el corolario 4.1, p_y toma el valor cero únicamente cuando $\lambda = 2$; en el resto de casos (es decir, para $\lambda > 2$), y teniendo en cuenta (2.26), se cumple $0 < p_y < 1$.

Por tanto, si el parámetro λ toma un valor mayor que 2, entonces el valor de los sumatorios $\sum_{j=0}^{t-1} p_y^j$ y $\sum_{j=1}^{t-1} p_y^j$ crece según aumenta el valor de t , fijados el resto de parámetros.

En consecuencia, aplicando esta propiedad a partir de la expresión (2.12) tenemos:

$$p_Q(t) = p_g^k \frac{\sum_{j=0}^{t-1} p_y^j}{1 - \sum_{j=1}^{t-1} p_y^j \sum_{i=1}^{k-1} p_g^i} < p_g^k \frac{\sum_{j=0}^t p_y^j}{1 - \sum_{j=1}^t p_y^j \sum_{i=1}^{k-1} p_g^i} = p_Q(t+1),$$

para todo $t \in \mathbb{Z}, t \geq 1$, como se quería demostrar.

Por último, la demostración de las igualdades (4.98) y (4.99) es inmediata a partir de las expresiones (2.13) y (2.12):

$$p_Q(t \mid k = 1 \text{ o } t = 1) = p_g^k \sum_{j=0}^{t-1} p_y^j$$

$$p_Q(t) = p_g^k \frac{\sum_{j=0}^{t-1} p_y^j}{1 - \sum_{j=1}^{t-1} p_y^j \sum_{i=1}^{k-1} p_g^i},$$

usando (2.15) $\sum_{i=0}^{+\infty} r^i = \frac{1}{1-r}, \forall r : |r| < 1$. □

Acabamos de probar que la probabilidad de cualificar el proceso crece y tiende asintóticamente a un cierto valor (menor que 1) cuando el número de piezas amarillas consecutivas t necesario para no cualificarlo tiende a infinito, fijados el resto de parámetros. Partiendo de este resultado se deducen de manera inmediata los siguientes:

Corolario 4.9 (Comportamiento de α como función de t). *Fijado el parámetro λ a un valor mayor que 2, la probabilidad de falsa alarma α disminuye según aumenta el valor de t , en todo el dominio de t , fijados arbitrariamente el*

resto de parámetros; es decir:

$$\alpha(t+1) < \alpha(t), \quad \text{para todo } t \in \mathbb{Z}, t \geq 1. \quad (4.101)$$

Además, en ese caso se verifica

$$\alpha(t=1) = 1 - p_g(\delta=0)^k, \quad (4.102)$$

$$\lim_{t \rightarrow +\infty} \alpha(t) = 1 - \frac{p_g(\delta=0)^k}{1 - p_y(\delta=0) \sum_{i=0}^{k-1} p_g(\delta=0)^i}. \quad (4.103)$$

Cuando $\lambda = 2$, la probabilidad α se mantiene constante respecto de t , fijados el resto de parámetros; esto es:

$$\alpha(t_1 | \lambda = 2) = \alpha(t_2 | \lambda = 2), \quad \forall t_1, t_2 \in \mathbb{Z}, t_1, t_2 \geq 1. \quad (4.104)$$

Demostración. La prueba es inmediata a partir de la proposición 4.4 que se acaba de demostrar y de la definición de α (2.20): $\alpha(k, t, \lambda | C_p) = 1 - p_Q(k, t, \lambda | C_p, \delta = 0)$. \square

Corolario 4.10 (Comportamiento de β como función de t). *Fijado el parámetro λ a un valor mayor que 2, la probabilidad β de no detectar una cierta desviación δ del proceso se incrementa según aumenta el valor de t , en todo el dominio de t , fijados arbitrariamente el resto de parámetros; es decir:*

$$\beta(t+1 | \delta) < \beta(t | \delta), \quad \text{para todo } t \in \mathbb{Z}, t \geq 1. \quad (4.105)$$

Además, en ese caso se verifica

$$\beta(t=1 | \delta) = p_g(\delta)^k, \quad (4.106)$$

$$\lim_{t \rightarrow +\infty} \beta(t | \delta) = \frac{p_g(\delta)^k}{1 - p_y(\delta) \sum_{i=0}^{k-1} p_g(\delta)^i}. \quad (4.107)$$

Cuando $\lambda = 2$, la probabilidad β se mantiene constante respecto de t , fijados el

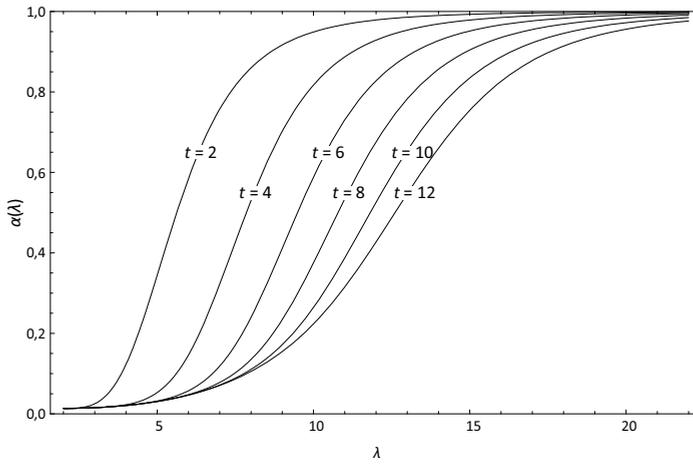


Figura 4.14: Comportamiento de α como función del parámetro t , en el precontrol con $k = 5$, para un proceso con capacidad potencial $C_p = 1$.

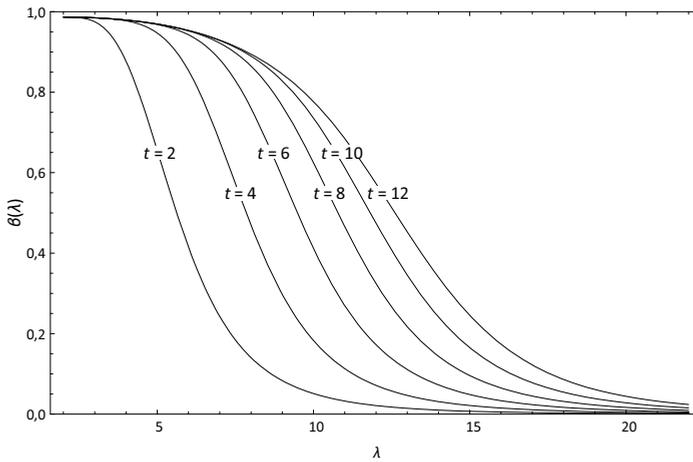


Figura 4.15: Comportamiento de $\beta(\cdot | \delta = 0)$ como función del parámetro t , en el precontrol con $k = 5$, para un proceso con capacidad potencial $C_p = 1$.

resto de parámetros; esto es:

$$\beta(t_1 | \lambda = 2, \delta) = \beta(t_2 | \lambda = 2, \delta), \quad \forall t_1, t_2 \in \mathbb{Z}, t_1, t_2 \geq 1. \quad (4.108)$$

Demostración. Se deduce de manera inmediata usando la proposición 4.4 y la definición de β (2.27): $\beta(k, t, \lambda | C_p, \delta) = p_Q(k, t, \lambda | C_p, \delta)$. \square

Las figuras 4.14 y 4.15 muestran el comportamiento de α y β como función de t que terminamos de probar.

Proposición 4.5 (Comportamiento de la función objetivo como función de t). *Fijados el resto de parámetros, el valor de la función objetivo del problema del precontrol óptimo (4.5) f se incrementa según aumenta el valor de t , si $\lambda > 2$, es decir:*

$$f(t+1 | \lambda = \lambda_0) > f(t | \lambda = \lambda_0), \quad \text{para todo } t \in \mathbb{Z}, t \geq 1, \text{ con } \lambda_0 > 2. \quad (4.109)$$

Para $\lambda = 2$ se cumple:

$$f(t_1 | \lambda = 2) = f(t_2 | \lambda = 2), \quad \text{para todo } t_1, t_2 \in \mathbb{Z}, t_1, t_2 \geq 1. \quad (4.110)$$

Además, en cualquier caso, se cumple:

$$f(t = 1) = \sum_{i=0}^{k-1} p_g(\delta = 0)^i, \quad (4.111)$$

$$\lim_{t \rightarrow +\infty} f(t) = \frac{1}{\frac{1-p_g(\delta=0)}{1-p_g(\delta=0)^k} - p_y(\delta=0)}. \quad (4.112)$$

Demostración. La prueba es similar a la de la proposición 4.3, intercambiando entre sí el papel de los parámetros k y t , y también el de las probabilidades p_g y p_y .

Únicamente es necesario detallar las diferencias entre los casos $\lambda = 2$ y $\lambda > 2$. La probabilidad de pieza amarilla p_y es nula si y sólo si $\lambda = 2$ (corolario 4.1). De ello, usando las expresiones conocidas para f (3.25) o (4.55), se deduce trivialmente (4.110).

Cuando $\lambda > 2$, sabemos que $p_y > 0$, lo cual implica $\sum_{j=0}^{t-1} p_y(\delta = 0)^j < \sum_{j=0}^t p_y(\delta = 0)^j$, y junto con la expresión (4.55) para f se tiene el resultado buscado. \square

La figura 4.16 muestra el comportamiento de f como función de t que se ha demostrado.

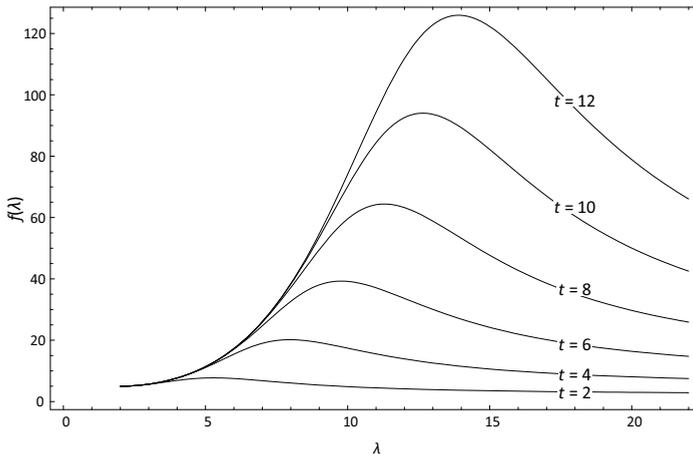


Figura 4.16: Comportamiento de f como función del parámetro t , en el precontrol con $k = 5$, para un proceso con capacidad potencial $C_p = 1$.

4.3.4. Comportamiento asintótico como función de k y t

¿Qué sucede con la probabilidad de cualificar p_Q y con el número medio esperado de piezas a muestrear f cuando el valor de los parámetros k y t varía simultáneamente? En general, el resultado dependerá del grado de variación de cada parámetro y del valor de las probabilidades p_g y p_y , entre otros factores. Pero sí puede caracterizarse el comportamiento asintótico de dichas funciones cuando k y t tienden a más infinito al mismo tiempo. Lo desarrollamos en las dos proposiciones siguientes.

Proposición 4.6 (Comportamiento asintótico de la probabilidad de cualificar como función de k y t). *Fijados el resto de parámetros, la probabilidad de cualificar p_Q , tomada como función de los parámetros k y t , presenta el siguiente comportamiento asintótico:*

$$\lim_{k,t \rightarrow +\infty} p_Q(k,t) = 0. \quad (4.113)$$

Demostración. La prueba es directa partiendo de los mismos elementos que los utilizados para la demostración de la expresión (4.82) en la proposición 4.2.

Otro modo de enfocar la demostración es tomar la expresión resultante de hacer tender t a más infinito en $p_Q(k,t)$ (4.99) y calcular su límite cuando k tiende a más infinito. \square

Lo interesante de este resultado es que el hecho de que t también tienda a infinito junto con k no impide que la probabilidad de cualificar termine acercándose a cero, tanto como se desee, para valores suficientemente grandes de k .

Corolario 4.11 (Comportamiento de α como función de k y t). *Fijados el resto de parámetros, la probabilidad de falsa alarma α presenta el siguiente comportamiento asintótico como función conjunta de k y t :*

$$\lim_{k,t \rightarrow +\infty} \alpha(k,t) = 1. \quad (4.114)$$

Demostración. El resultado es inmediato usando la proposición 4.6 que acabamos de demostrar, junto con la definición de α (2.20). \square

Corolario 4.12 (Comportamiento de β como función de k y t). *Fijados el resto de parámetros, la probabilidad β de no detectar una cierta desviación δ del proceso presenta el siguiente comportamiento asintótico como función conjunta de k y t :*

$$\lim_{k,t \rightarrow +\infty} \beta(k,t | \delta) = 0. \quad (4.115)$$

Demostración. Es inmediato usando la proposición 4.6 y la definición de β (2.27). \square

Proposición 4.7 (Comportamiento asintótico de la función objetivo como función de k y t). *Fijados el resto de parámetros, la función objetivo del problema del precontrol óptimo (4.5) f , tomada como función de los parámetros k y t , presenta el siguiente comportamiento asintótico:*

$$\lim_{k,t \rightarrow +\infty} f(k,t) = \frac{1}{p_r(\delta = 0)}. \quad (4.116)$$

Demostración. Se puede obtener la igualdad aplicando el doble límite a la expresión para f (4.55):

$$f(k,t) = \frac{1}{\left(\sum_{i=0}^{k-1} p_g(\delta = 0)^i\right)^{-1} + \left(\sum_{j=0}^{t-1} p_y(\delta = 0)^j\right)^{-1} - 1},$$

usando (2.15) $\sum_{i=0}^{+\infty} r^i = \frac{1}{1-r}$, $\forall r : |r| < 1$ y (3.4) $p_r = 1 - p_g - p_y$. \square

El resultado anterior es claro: cuando el número de piezas consecutivas verdes y amarillas necesario para, respectivamente, cualificar y no cualificar el proceso es tan elevado que su observación es prácticamente imposible, casi la única posibilidad de terminar el procedimiento del precontrol es obtener una pieza roja. Precisamente, el valor esperado en tal caso para f , según la proposición que acabamos de demostrar, tiende al valor de la esperanza matemática del número de piezas a muestrear hasta obtener una roja (variable aleatoria con distribución geométrica de parámetro $p_r(\delta = 0)$).

4.4. Optimización del valor de λ , fijados k y t

Los resultados demostrados previamente son interesantes en sí mismos porque confirman aspectos básicos del comportamiento del precontrol generalizado; pero su utilidad se muestra sobre todo a partir de

este punto, en que comenzamos a utilizar lo que conocemos y hemos demostrado acerca de las funciones α , β y f para avanzar en el estudio del problema OPC (4.5) y en el diseño de algoritmos específicos para su resolución, que es el objetivo último.

Concretamente, en la presente sección generaremos la base teórica de una metodología, que será desarrollada en el siguiente capítulo, dirigida a determinar el valor de la variable λ que minimiza el valor de la función objetivo f , respetando las restricciones y estando fijados arbitrariamente los valores de k y t .

Esta estrategia parece a priori interesante, dado que, en caso de obtener una manera sistemática y, sobre todo, *fiabile* de determinar el valor óptimo de λ en tal situación, podría implementarse y ser integrada como parte de algoritmos de resolución que tratarían con soluciones de la forma (k, t) ; el valor de λ se *optimizaría* separadamente para cada combinación de k y t .

Por tanto, en esta sección nos detendremos en estudiar y resolver de manera analítica el que llamaremos *problema OPC-NLP*($k, t, \alpha^*, \beta^*, \delta^*$), problema OPC-NLP(k, t) o simplemente problema OPC-NLP, resultante de fijar arbitrariamente los valores de k y t en el problema OPC (4.5), y que puede expresarse de la siguiente forma:

$$\left. \begin{array}{ll} \text{Min} & f(\lambda | k, t, C_p) \\ \text{s.a:} & \alpha(\lambda | k, t, C_p) \leq \alpha^* \\ & \beta(\lambda | k, t, C_p, \delta^*) \leq \beta^* \\ & \lambda \geq 2 \end{array} \right\}, \quad (4.117)$$

donde los parámetros del precontrol k y t se consideran fijados a sendos valores enteros mayores o iguales a 1, constantes y conocidos, y los parámetros C_p , α^* , β^* , δ^* mantienen el mismo significado que en el problema del precontrol óptimo (4.5). Es decir, el modelo OPC-NLP (4.117) se trata de un problema de optimización no lineal restringida univariante.

Posteriormente, extraeremos algunas conclusiones aplicables a la resolución del problema OPC (4.5).

4.4.1. Optimización local de λ

En primer lugar, deduciremos cuál es el rango de valores de la variable λ que hacen que se verifiquen las restricciones del problema OPC-NLP (4.117), y posteriormente analizaremos cuál de esos valores es el que optimiza el valor de la función objetivo f .

A continuación se caracteriza el rango de valores de λ que verifican la primera restricción del problema OPC-NLP (4.117).

Proposición 4.8 (Definición de λ_α). *Dados $k_0, t_0 \geq 1$ enteros y $\alpha^* \in]0, 1[$, se cumple*

$$\begin{aligned} & \alpha(\lambda = 2 \mid k = k_0, t = 1) \leq \alpha^* \\ \Leftrightarrow & \exists \lambda_\alpha \geq 2 \text{ tal que } \alpha(\lambda = \lambda_\alpha \mid k = k_0, t = t_0) = \alpha^*; \end{aligned} \quad (4.118)$$

en caso de existir, el valor λ_α es único, y se verifica

$$\Lambda_\alpha = \{ \lambda \geq 2 \mid \alpha(\lambda \mid k_0, t_0, C_p) \leq \alpha^* \} = [2, \lambda_\alpha]. \quad (4.119)$$

En caso de no existir λ_α , se cumple

$$\Lambda_\alpha = \{ \lambda \geq 2 \mid \alpha(\lambda \mid k_0, t_0, C_p) \leq \alpha^* \} = \emptyset. \quad (4.120)$$

Demostración. En primer lugar probaremos la implicación de izquierda a derecha de la doble implicación (4.118).

Si $\alpha(\lambda = 2 \mid k = k_0, t = 1) \leq \alpha^*$, entonces por la igualdad (4.104) demostrada en el corolario 4.9 se cumple:

$$\alpha(\lambda = 2 \mid k = k_0, t = t_0) \leq \alpha^*. \quad (4.121)$$

Por el corolario 4.2 sabemos que α es continua (y estrictamente creciente) como función de λ en $[2, +\infty[$, fijadas k y t , y que $\lim_{\lambda \rightarrow +\infty} \alpha(\lambda) = 1$, según (4.34). Teniendo en cuenta además (4.121) y $\alpha^* < 1$, por el teorema de los valores intermedios podemos concluir que

$$\exists \lambda_\alpha \geq 2 \text{ tal que } \alpha(\lambda_\alpha \mid k_0, t_0, C_p) = \alpha^*, \quad (4.122)$$

como se quería demostrar. Además, del hecho de que $\alpha(\lambda)$ es continua y estrictamente creciente se deduce que sólo existe un valor $\lambda_\alpha \geq 2$ que verifique (4.122), y que en caso de existir necesariamente se cumple

$$\begin{aligned}\alpha(\lambda \mid k_0, t_0, C_p) &\leq \alpha^* \quad \forall \lambda \leq \lambda_\alpha, \lambda \geq 2 \quad \text{y} \\ \alpha(\lambda \mid k_0, t_0, C_p) &> \alpha^* \quad \forall \lambda > \lambda_\alpha,\end{aligned}$$

con lo que quedaría demostrada la expresión (4.119).

En particular, (4.119) se cumplirá para $\lambda = 2$, es decir, $\alpha(\lambda = 2 \mid k = k_0, t = t_0) \leq \alpha^*$; y de nuevo por (4.104) se tiene

$$\alpha(\lambda = 2 \mid k = k_0, t = 1) \leq \alpha^*,$$

lo cual prueba la implicación de derecha a izquierda de (4.118).

Por último, supongamos que estamos en el caso de que no existe un valor λ_α tal que cumpla $\alpha(\lambda_\alpha \mid k_0, t_0, C_p) = \alpha^*$; según (4.118) esto es así si y sólo si $\alpha(\lambda = 2 \mid k = k_0, t = 1) > \alpha^*$. Por (4.104) se cumple:

$$\alpha(\lambda = 2 \mid k = k_0, t = t_0) > \alpha^*. \quad (4.123)$$

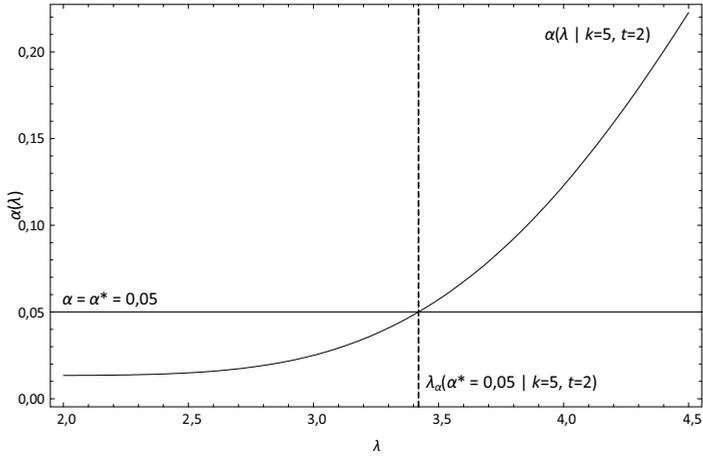
Aplicando de nuevo que $\alpha(\lambda)$ es continua y estrictamente creciente en todo su dominio, a partir de (4.123) se deduce

$$\alpha(\lambda \mid k_0, t_0, C_p) > \alpha^* \quad \forall \lambda \geq 2,$$

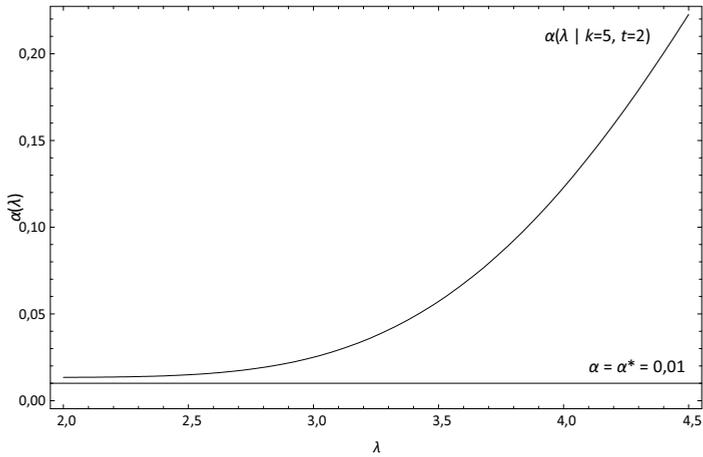
y con ello que queda demostrada la igualdad (4.120). \square

La proposición anterior proporciona una condición necesaria y suficiente para la existencia de un conjunto de valores de λ que verifiquen la primera restricción del problema OPC-NLP(k, t) (4.117). La importancia de este resultado está principalmente en que demuestra que la existencia de λ_α es independiente del valor que toma el parámetro t ; depende únicamente del valor de k .

En la figura 4.17 se muestra, a través de dos ejemplos, la determinación del valor de λ_α para una configuración concreta de k, t y α^* .



(a)



(b)

Figura 4.17: Determinación de λ_α en dos supuestos diferentes, ambos con $k = 5$ y $t = 2$, en un proceso con capacidad potencial $C_p = 1$. La figura (a) muestra la determinación del valor de λ_α asociado a un valor de α^* igual a 0,05. La figura (b) ilustra el caso $\alpha(\lambda) > \alpha^* \forall \lambda \geq 2$, en que no existiría λ_α y ningún valor de λ verificaría la primera restricción del problema OPC-NLP (4.117) para la combinación de k y t considerada.

A continuación demostramos un resultado similar para la segunda restricción del problema OPC-NLP (4.117).

Proposición 4.9 (Definición de λ_β). *Dados $k_0, t_0 \geq 1$ enteros, $\delta^* > 0$ y $\beta^* \in]0, 1[$, existe siempre un único valor $\lambda_\beta \geq 2$ tal que*

$$\Lambda_\beta = \{ \lambda \geq 2 \mid \beta(\lambda \mid k_0, t_0, C_p, \delta^*) \leq \beta^* \} = [\lambda_\beta, +\infty[. \quad (4.124)$$

Concretamente, se cumple

$$\begin{aligned} \lambda_\beta &= 2 && \text{si } S(C_p, \delta^*)^{k_0} < \beta^*, \\ \beta(\lambda_\beta \mid k_0, t_0, C_p, \delta^*) &= \beta^* && \text{si } S(C_p, \delta^*)^{k_0} \geq \beta^*. \end{aligned} \quad (4.125)$$

Demostración. De manera análoga a lo demostrado para $\alpha(\lambda)$ y λ_α , la prueba se basa principalmente en la continuidad y la monotonía de la función $\beta(\lambda)$ demostrada en el corolario 4.3.

Según la igualdad (4.35) demostrada en dicho corolario,

$$\beta(\lambda = 2 \mid k_0, t_0, C_p, \delta^*) = S(C_p, \delta^*)^{k_0},$$

donde S (4.1) representa la probabilidad de que una observación esté dentro de especificaciones.

En el caso en que $S(C_p, \delta^*)^{k_0} < \beta^*$, por (4.35),

$$\beta(\lambda = 2 \mid k_0, t_0, C_p, \delta^*) < \beta^*,$$

y por ser $\beta(\lambda)$ continua y estrictamente decreciente en todo el dominio de λ , se tiene

$$\beta(\lambda \mid k_0, t_0, C_p, \delta^*) < \beta^* \quad \forall \lambda \geq 2.$$

Es decir, si $S(C_p, \delta^*)^{k_0} < \beta^*$, entonces

$$\{ \lambda \geq 2 \mid \beta(\lambda \mid k_0, t_0, C_p, \delta^*) \leq \beta^* \} = [2, +\infty[;$$

por tanto existe un único λ_β que cumple la definición (4.124), y en concreto toma el valor 2 (esto es, todo el rango de valores de λ cumple la restricción

segunda del problema OPC-NLP (4.117); la restricción es redundante). Con todo ello hemos probado la expresión (4.125) en su primer caso.

Ahora, si $S(C_p, \delta^*)^{k_0} \geq \beta^*$, teniendo en cuenta de nuevo (4.35), llegamos a

$$\beta(\lambda = 2 \mid k_0, t_0, C_p, \delta^*) \geq \beta^*. \quad (4.126)$$

Usando que $\beta(\lambda)$ es continua, que $\lim_{\lambda \rightarrow +\infty} \beta(\lambda \mid \delta) = 0$ según (4.36) (demostrado en el corolario 4.3) y la expresión (4.126), y teniendo en cuenta que $\beta^* > 0$, por el teorema de los valores intermedios tenemos que

$$\exists \lambda_\beta \geq 2 \text{ tal que } \beta(\lambda_\beta \mid k_0, t_0, C_p, \delta^*) = \beta^*. \quad (4.127)$$

Como consecuencia de que $\beta(\lambda)$ es estrictamente decreciente, necesariamente λ_β es único, y además

$$\begin{aligned} \beta(\lambda \mid k_0, t_0, C_p, \delta^*) &> \beta^* & \forall \lambda < \lambda_\beta, \lambda \geq 2 \text{ y} \\ \beta(\lambda \mid k_0, t_0, C_p, \delta^*) &\leq \beta^* & \forall \lambda \geq \lambda_\beta, \end{aligned}$$

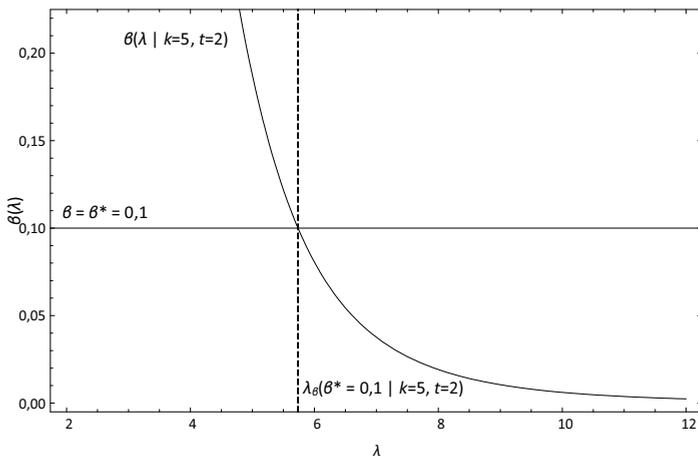


Figura 4.18: Determinación de λ_β para $\beta^* = 0,1$, $\delta^* = 0,9454$ (tal que $p^* = 0,02$; véase (2.34)), $k = 5$ y $t = 2$, en un proceso con capacidad potencial $C_p = 1$.

es decir, λ_β cumple (4.124)

$$\{ \lambda \geq 2 \mid \beta(\lambda \mid k_0, t_0, C_p, \delta^*) \leq \beta^* \} = [\lambda_\beta, +\infty[,$$

y por (4.127) podemos concluir que hemos demostrado el segundo caso de la expresión (4.125). \square

En la figura 4.18 se presenta un ejemplo de determinación de λ_β para una configuración concreta de los parámetros k , t y β^* .

Estamos en condiciones de probar el siguiente teorema, en el cual unimos todo lo ya demostrado sobre la región factible del problema no lineal univariante OPC-NLP (4.117) y caracterizamos de manera constructiva la existencia de solución óptima, en general única.

Teorema 4.3 (Optimización local de λ). *Fijados $k_0, t_0 \in \mathbb{Z}$, $k_0, t_0 \geq 1$, $\alpha^* \in]0, 1[$, $\beta^* \in]0, 1[$ y $\delta^* > 0$ el problema de optimización no lineal restringida univariante OPC-NLP (4.117)*

$$\left. \begin{array}{l} \text{Min} \quad f(\lambda \mid k_0, t_0, C_p) \\ \text{s.a:} \quad \alpha(\lambda \mid k_0, t_0, C_p) \leq \alpha^* \\ \quad \quad \beta(\lambda \mid k_0, t_0, C_p, \delta^*) \leq \beta^* \\ \quad \quad \lambda \geq 2 \end{array} \right\} ,$$

definido sobre un proceso con capacidad potencial conocida $C_p > 0$, tiene solución finita si y sólo si

$$\exists \lambda_\alpha \quad \text{y} \quad \lambda_\beta \leq \lambda_\alpha \tag{4.128}$$

siendo λ_α y λ_β tales que

$$\begin{aligned} \alpha(\lambda_\alpha \mid k_0, t_0, C_p) &= \alpha^* , \\ \lambda_\beta &= 2 \quad \text{si } S(C_p, \delta^*)^{k_0} < \beta^* , \\ \beta(\lambda_\beta \mid k_0, t_0, C_p, \delta^*) &= \beta^* \quad \text{si } S(C_p, \delta^*)^{k_0} \geq \beta^* . \end{aligned} \tag{4.129}$$

En tal caso, el conjunto de soluciones posibles para (4.117) es

$$\Lambda = [\lambda_\beta, \lambda_\alpha], \quad (4.130)$$

y el valor óptimo de λ , al que nos referiremos como λ' , se alcanza en uno de los límites del intervalo, es decir:

$$\lambda' = \begin{cases} \lambda_\alpha & \text{si } f(\lambda_\alpha) < f(\lambda_\beta) \\ \lambda_\beta & \text{en caso contrario.} \end{cases} \quad (4.131)$$

En otro caso (no existencia de λ_α o bien $\lambda_\beta > \lambda_\alpha$), el problema (4.117) es imposible.

Una condición necesaria y suficiente para asegurar la existencia de λ_α es que se verifique la siguiente desigualdad:

$$\alpha(\lambda = 2 \mid k = k_0, t = 1) \leq \alpha^*.$$

Demostración. Parte del enunciado de este teorema se ha demostrado ya en las proposiciones 4.8 y 4.9. Según la proposición 4.8, el conjunto de valores $\lambda \geq 2$ que verifican la primera restricción del problema (4.117) es no vacío si y sólo si existe $\lambda_\alpha \geq 2$ tal que $\alpha(\lambda_\alpha \mid k_0, t_0, C_p) = \alpha^*$, y dicho λ_α existe si y sólo si $\alpha(\lambda = 2 \mid k = k_0, t = 1) \leq \alpha^*$, como queríamos demostrar.

Además, en caso de existir λ_α , la proposición 4.8 establece que el conjunto de valores de λ que cumplen dicha restricción es (4.119)

$$\Lambda_\alpha = \{ \lambda \geq 2 \mid \alpha(\lambda \mid k_0, t_0, C_p) \leq \alpha^* \} = [2, \lambda_\alpha].$$

Por otro lado, en la proposición 4.9 se ha probado que el conjunto de valores $\lambda \geq 2$ que cumplen la segunda restricción del problema (4.117) es (4.124)

$$\Lambda_\beta = \{ \lambda \geq 2 \mid \beta(\lambda \mid k_0, t_0, C_p, \delta^*) \leq \beta^* \} = [\lambda_\beta, +\infty[,$$

donde λ_β cumple

$$\begin{aligned} \lambda_\beta &= 2 && \text{si } S(C_p, \delta^*)^{k_0} < \beta^*, \\ \beta(\lambda_\beta \mid k_0, t_0, C_p, \delta^*) &= \beta^* && \text{si } S(C_p, \delta^*)^{k_0} \geq \beta^*. \end{aligned}$$

Llamemos Λ a la región factible del problema (4.117), es decir, al conjunto de valores de λ que verifican simultáneamente todas y cada una de las restricciones del problema. Por como se han definido los conjuntos Λ_α , Λ_β y Λ , claramente se cumple

$$\Lambda = \Lambda_\alpha \cap \Lambda_\beta. \quad (4.132)$$

Si no existe λ_α , sabemos que $\Lambda_\alpha = \emptyset$ y por tanto, por (4.132), también $\Lambda = \emptyset$; es decir, el problema (4.117) es imposible en ese caso, debido a que ningún valor de $\lambda \geq 2$ cumple la restricción acerca de la tasa de falsa alarma.

En caso de existir λ_α , a partir de (4.119), (4.124) y (4.132) se deduce que

$$\Lambda = \begin{cases} \emptyset & \text{si } \lambda_\beta > \lambda_\alpha \\ [\lambda_\beta, \lambda_\alpha] & \text{si } \lambda_\beta \leq \lambda_\alpha. \end{cases}$$

Es decir, en el caso en que $\lambda_\beta > \lambda_\alpha$, el problema (4.117) será imposible debido a que no habrá ningún valor de $\lambda \geq 2$ que cumpla al mismo tiempo las restricciones acerca de las probabilidades α y β .

En otro caso, el problema tendrá como región factible el intervalo cerrado y acotado $[\lambda_\beta, \lambda_\alpha] \subseteq [2, +\infty[$, como también queríamos demostrar. Por ser la función objetivo $f(\lambda)$ continua en $[2, +\infty[$ según el teorema 4.2, necesariamente existirá al menos un punto en $[\lambda_\beta, \lambda_\alpha]$ donde se alcanzará el mínimo de $f(\lambda)$ (teorema de Weierstrass o de los valores extremos), es decir, el problema tendrá solución óptima finita. Y con ello hemos probado todo lo relativo a la existencia de soluciones del problema OPC-NLP (4.117) en el enunciado de este teorema.

En este último caso en que existe solución (es decir, cuando se cumple (4.128)), vamos a deducir cuál es el valor del intervalo $[\lambda_\beta, \lambda_\alpha]$ en el que se

alcanza el mínimo de la función objetivo del problema OPC-NLP (4.117).

Cuando se trata de optimizar el valor de una función continua (y derivable) como es $f(\lambda)$ en un dominio cerrado y acotado como es $[\lambda_\beta, \lambda_\alpha]$, pueden darse dos casos (no excluyentes): que el óptimo se alcance en el interior del conjunto (en nuestro caso, en el intervalo abierto $] \lambda_\beta, \lambda_\alpha [$), o bien que el óptimo se alcance en la frontera del dominio (es decir, en los extremos λ_β y λ_α).

Por el corolario 4.5, sabemos que, si $(k, t) \neq (1, 1)$, $f(\lambda)$ no presenta ningún mínimo local en todo $]2, +\infty[$, y por tanto no lo hay en el interior de $[\lambda_\beta, \lambda_\alpha]$. En consecuencia, la única posibilidad es que el mínimo de f en $[\lambda_\beta, \lambda_\alpha]$ se alcance en los extremos del intervalo, esto es, en λ_β o λ_α .

En el caso $(k, t) = (1, 1)$, según el corolario 4.4, f es constante como función de λ , y por tanto el mínimo de la función se alcanza en cualquier punto del dominio $[\lambda_\beta, \lambda_\alpha]$; en particular, se alcanzará también en los extremos del intervalo.

En conclusión, si, en caso de existir, denotamos por λ' a la solución óptima del problema OPC-NLP (4.117), es decir, al valor de λ que minimiza el valor de f en el dominio Λ , entonces, según acabamos de deducir:

$$\lambda' = \begin{cases} \lambda_\alpha & \text{si } f(\lambda_\alpha) < f(\lambda_\beta) \\ \lambda_\beta & \text{en caso contrario,} \end{cases}$$

como se quería demostrar. \square

El resultado que acabamos de probar, que se ilustra en la figura 4.19 a través de un ejemplo concreto, permite *optimizar* de manera separada el valor de la única variable continua involucrada en el modelo OPC (4.5), y obtener así lo que podría llamarse un *óptimo local* de la función objetivo para cada combinación de valores de las variables de carácter discreto k y t .

Lo particularmente importante que se deriva de este resultado es que, por las características de la función objetivo y de las restricciones del problema OPC-NLP (4.117), el óptimo para λ , fijados k y t , se encuentra en los límites del *intervalo de factibilidad*, esto es, en lo que hemos llamado

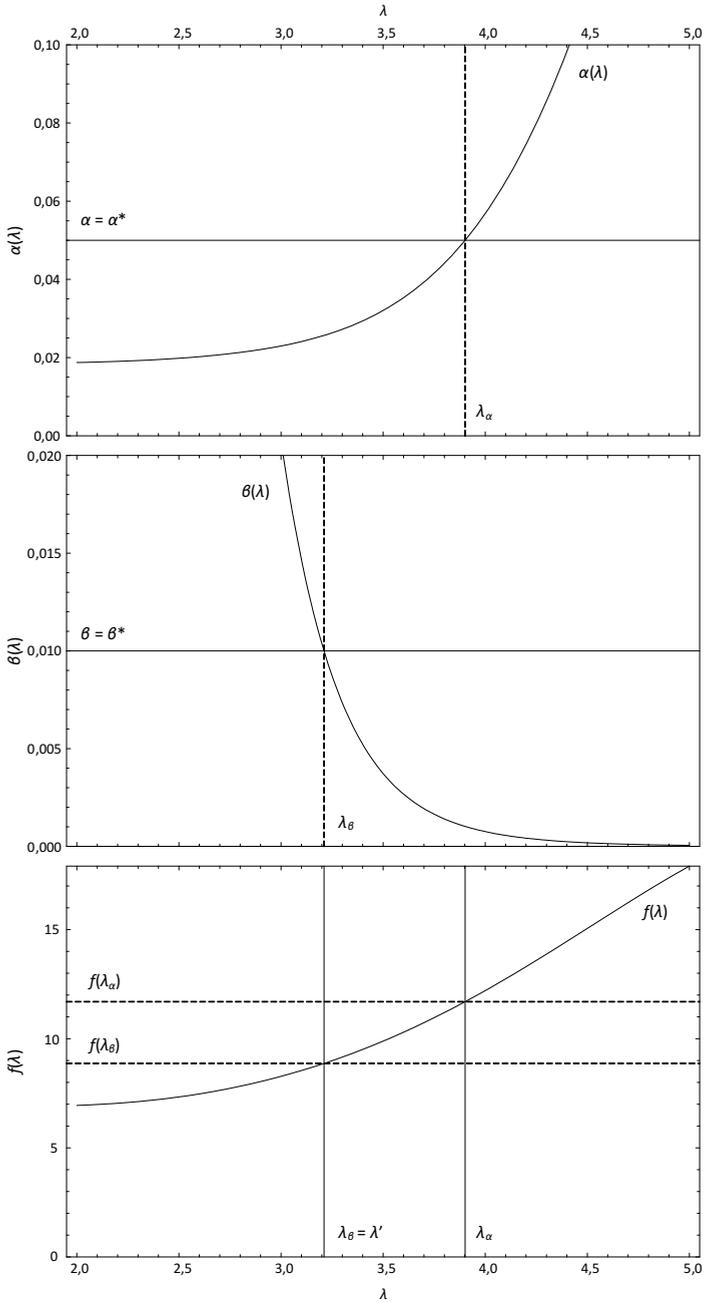


Figura 4.19: Determinación de λ' para $\alpha^* = 0,05$, $\beta^* = 0,01$, $\delta^* = 2$, $k = 7$ y $t = 3$, en un proceso con capacidad potencial $C_p = 1$. En primer lugar, se obtienen λ_α y λ_β . Posteriormente, se calcula el valor de f para ambos valores de λ , y aquel que proporcione un menor valor para la función objetivo es λ' , para la combinación de k y t considerada.

λ_α o en λ_β (véase la expresión (4.131)), que son los respectivos valores de λ para los cuales se cumplen con igualdad las dos restricciones del problema OPC-NLP (4.117) (excepto, en algún caso, λ_β ; véase (4.125)). Por tanto, la búsqueda del valor óptimo de λ se reduce a la resolución de dos ecuaciones (no lineales) o, equivalentemente, a la búsqueda de las raíces de dos funciones. En definitiva, para *optimizar* λ no estamos obligados a aplicar ninguna técnica de optimización continua, ya que hay un camino más directo consistente en utilizar técnicas de búsquedas de raíces (*root finding methods*, en inglés).

Además, por la relación entre los problemas OPC-NLP (4.117) y OPC (4.5), el teorema 4.3 proporciona un criterio directo para determinar la *factibilidad* o *infactibilidad* de una solución dada (k, t) , independientemente del valor de la variable λ . Concretamente, se ha visto que una solución (k, t) puede ser no factible en el problema del precontrol óptimo (4.5), para cualquier valor de λ , debido a uno de los siguientes dos motivos:

Infactibilidad por α Fijados los valores de k y t , no hay ningún valor de λ que verifique la restricción primera del OPC (4.5) $\alpha(k, t, \lambda) \leq \alpha^*$. Es el caso en el que no existe λ_α . Además, se ha demostrado que este tipo de infactibilidad depende únicamente del valor de k .

Infactibilidad por $\lambda_\alpha < \lambda_\beta$ Fijados los valores de k y t , *sí* hay valores de λ que verifican la primera ecuación del problema (4.5), pero *no* coinciden con los valores de λ que cumplen la segunda restricción $\beta(k, t, \lambda | \delta^*) \leq \beta^*$. Siguiendo la notación empleada en el teorema, es el caso en que $\Lambda_\alpha \neq \emptyset$ pero $\Lambda_\alpha \cap \Lambda_\beta = \emptyset$.

Nótese que *no* tendría sentido hablar de *infactibilidad por β* , ya que, fijadas k y t , siempre hay un conjunto de valores $[\lambda_\beta, +\infty[$ que verifican la restricción segunda del problema OPC, tal y como se ha demostrado en la proposición 4.9.

Como ya se ha dicho, este resultado hace posible la estrategia de búsqueda de soluciones empleada en el algoritmo que hemos diseñado para resolver el problema (4.7), y que describiremos en el siguiente capítulo.

4.4.2. Consecuencias en la resolución del problema del precontrol óptimo

A continuación demostramos un resultado que se deduce directamente del teorema 4.3, y que, junto con este, será utilizado como base para el algoritmo que propondremos.

Corolario 4.13 (Optimización del precontrol con cotas superiores para k y t). *Fijados $K_{\text{máx}}, T_{\text{máx}} \in \mathbb{Z}$, $K_{\text{máx}}, T_{\text{máx}} \geq 1$, $\alpha^* \in]0, 1[$, $\beta^* \in]0, 1[$ y $\delta^* > 0$, y conocido el valor del parámetro $C_p > 0$, el problema de optimización no lineal entera mixta OPC-sb (4.7), definido como*

$$\left. \begin{array}{l} \text{Min} \quad f(k, t, \lambda \mid C_p) \\ \text{s.a:} \quad \alpha(k, t, \lambda \mid C_p) \leq \alpha^* \\ \quad \quad \beta(k, t, \lambda \mid C_p, \delta^*) \leq \beta^* \\ \quad \quad \quad 1 \leq k \leq K_{\text{máx}} \\ \quad \quad \quad 1 \leq t \leq T_{\text{máx}} \\ \quad \quad \quad \lambda \geq 2 \\ \quad \quad k, t \in \mathbb{Z}, \lambda \in \mathbb{R} \end{array} \right\},$$

tiene como región factible el conjunto

$$\text{RF}_{\text{OPCSb}} = \bigcup_{\substack{k=1, \dots, K_{\text{máx}} \\ t=1, \dots, T_{\text{máx}}}} \{ (k, t, \lambda) \mid \lambda \in \Lambda(k, t) \}, \quad (4.133)$$

donde $\Lambda(k, t)$ se define para cada par (k, t) según (4.132).

En el caso en que RF_{OPCSb} sea no vacío, el problema OPC-sb tiene solución óptima finita (k^*, t^*, λ^*) , siendo

$$f(k^*, t^*, \lambda^*) = \min_{\substack{k=1, \dots, K_{\text{máx}} \\ t=1, \dots, T_{\text{máx}}}} f(k, t, \lambda'(k, t)), \quad (4.134)$$

donde $\lambda'(k, t)$, en caso de existir, se define para cada par (k, t) según (4.131).

Demostración. La prueba es inmediata a partir del teorema 4.3. En él se demuestra que, para cada par (k, t) , con $k, t \in \mathbb{Z}$ y $k, t \geq 1$, el problema OPC-NLP(k, t) (4.117), cuya única variable de decisión es λ , o bien es imposible o bien tiene como región factible a $\Lambda(k, t)$ (4.132) y como solución óptima a $\lambda = \lambda'(k, t)$ (4.131).

Por el modo como se han definido ambos problemas, es claro que toda solución factible λ de OPC-NLP(k, t) (4.117), con $1 \leq k \leq K_{\text{máx}}$ y $1 \leq t \leq T_{\text{máx}}$, lo es si y sólo si (k, t, λ) pertenece a la región factible de OPC-sb (4.7). Esto implica directamente

$$\text{RF}_{\text{OPCsb}} = \bigcup_{\substack{k=1, \dots, K_{\text{máx}} \\ t=1, \dots, T_{\text{máx}}}} \{ (k, t, \lambda) \mid \lambda \in \Lambda(k, t) \},$$

como se quería demostrar.

Supongamos ahora que la región factible RF_{OPCsb} del problema OPC-sb (4.7) es no vacía. Dado que, como acabamos de probar, dicha región puede expresarse como unión (disjunta) de una serie de subconjuntos, buscar el óptimo (k^*, t^*, λ^*) en RF_{OPCsb} se reduce a buscarlo dentro de cada uno de los subconjuntos de la forma $\{ (k, t, \lambda) \mid \lambda \in \Lambda(k, t) \}$, con $k, t \in \mathbb{Z}$, $1 \leq k \leq K_{\text{máx}}$ y $1 \leq t \leq T_{\text{máx}}$; es decir:

$$\begin{aligned} f(k^*, t^*, \lambda^*) &= \min_{(k,t,\lambda) \in \text{RF}_{\text{OPCsb}}} f(k, t, \lambda) \\ \Leftrightarrow f(k^*, t^*, \lambda^*) &= \min_{\substack{k=1, \dots, K_{\text{máx}} \\ t=1, \dots, T_{\text{máx}}}} \left(\min_{\lambda \in \Lambda(k,t)} f(k, t, \lambda) \right). \end{aligned} \quad (4.135)$$

Estamos suponiendo $\text{RF}_{\text{OPCsb}} \neq \emptyset$; al tratarse de la unión de conjuntos de la forma $\{ (k, t, \lambda) \mid \lambda \in \Lambda(k, t) \}$, necesariamente existirá algún par (k, t) tal que $\Lambda(k, t) \neq \emptyset$. Ahora bien, fijadas k y t , sabemos por el teorema 4.3 que, en el caso de que $\Lambda(k, t) \neq \emptyset$, $\min \{ f(k, t, \lambda) \mid \lambda \in \Lambda(k, t) \} =$

$f(k, t, \lambda'(k, t))$. Uniendo esto a (4.135) tenemos

$$f(k^*, t^*, \lambda^*) = \min_{\substack{k = 1, \dots, K_{\text{máx}} \\ t = 1, \dots, T_{\text{máx}}}} f(k, t, \lambda'(k, t)),$$

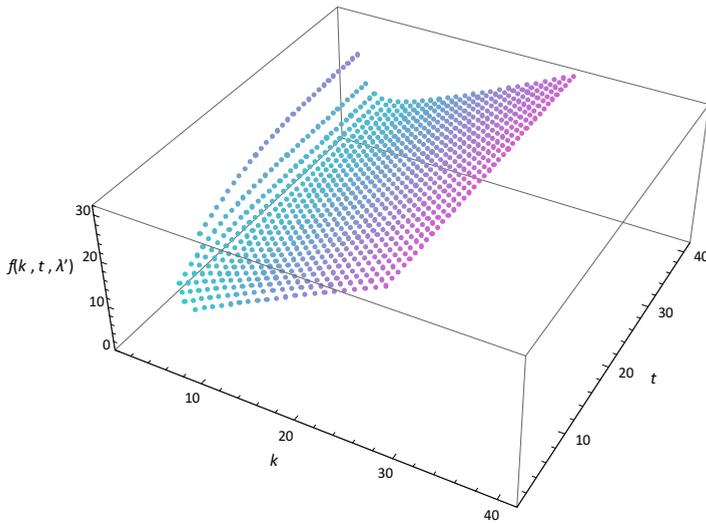
como también se quería demostrar. \square

Lo que se acaba de probar es una consecuencia lógica del teorema 4.3: en caso de existir, la solución óptima del problema OPC-sb (4.7) (que es el resultado de acotar superiormente el valor de las variables k y t en el problema OPC (4.5)) se puede obtener como la mejor de las soluciones óptimas *locales* asociadas a cada uno de los subproblemas OPC-NLP(k, t) en que se puede descomponer el problema original.

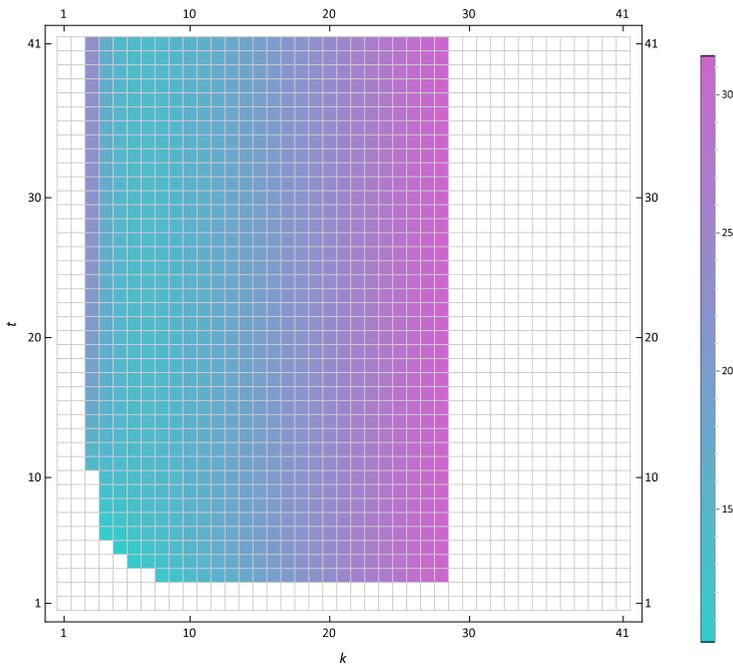
Como resulta evidente, este corolario proporciona la base teórica para una estrategia de búsqueda enumerativa, la cual es utilizada en el algoritmo que hemos desarrollado y que presentaremos en el capítulo 5.

Dada la importancia de este resultado, se muestran a continuación dos ejemplos de su aplicación a sendos casos concretos. En el primer ejemplo (figura 4.20) se resuelve el problema OPC-sb($\alpha^* = 0,01, \beta^* = 0,1, \delta^* = 1,5462, K_{\text{máx}} = 41, T_{\text{máx}} = 41$) para un proceso con $C_p = 1,2$, y en el segundo (figura 4.21) se trata el problema OPC-sb($\alpha^* = 0,005, \beta^* = 0,005, \delta^* = 1,9463, K_{\text{máx}} = 41, T_{\text{máx}} = 41$) para un proceso con $C_p = 1,3333$. Ambos problemas pertenecen a la batería de ejemplos que fue objeto de estudio en la sección 3.5.2 (ejemplos 9 y 13, respectivamente).

Para cerrar el estudio formal del problema OPC (4.5), ampliamos ahora lo tratado en la proposición 4.8 y en el teorema 4.3 en relación a la llamada *infectibilidad por α* .

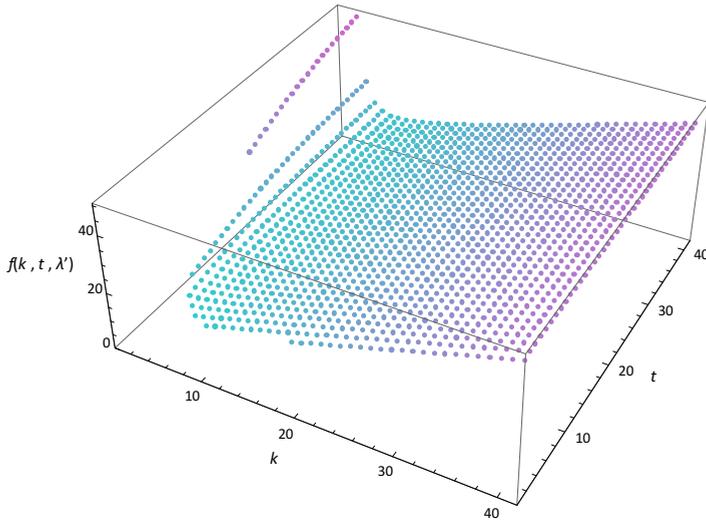


(a) Representación tridimensional.

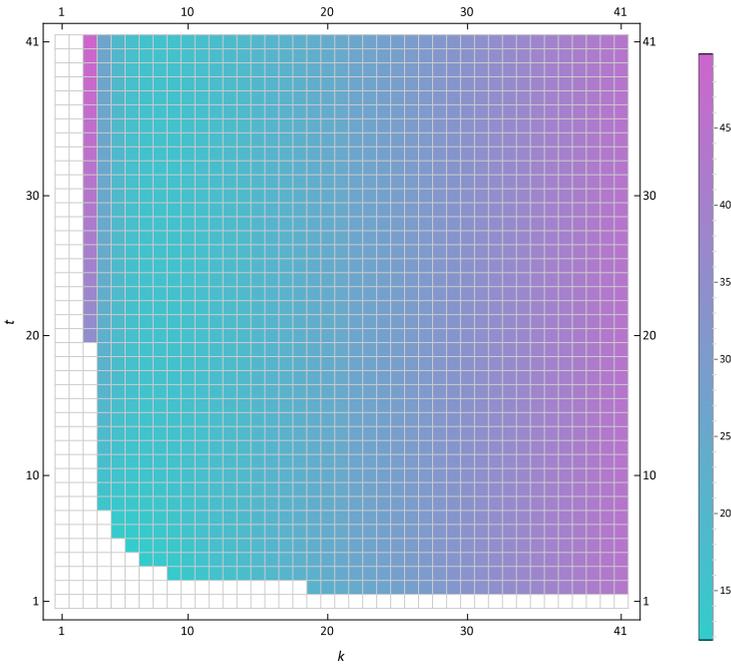


(b) Mapa de calor.

Figura 4.20: Resolución del problema OPC-sb (4.7), con parámetros $C_p = 1,2$, $\alpha^* = 0,01$, $\beta^* = 0,1$, $\delta^* = 1,5462$, $K_{\max} = 41$ y $T_{\max} = 41$. Se observa cómo a partir de $k = 29$ no existen soluciones factibles (en concreto, a partir de $k = 32$ se da el caso de infactibilidad por α ; véanse más adelante el corolario 4.14 y el ejemplo desarrollado en la subsección 5.4.5).



(a) Representación tridimensional.



(b) Mapa de calor.

Figura 4.21: Resolución del problema OPC-sb (4.7), con parámetros $C_p = 1,333$, $\alpha^* = 0,005$, $\beta^* = 0,005$, $\delta^* = 1,9463$, $K_{\text{máx}} = 41$ y $T_{\text{máx}} = 41$.

Proposición 4.10. *Dados $k_0 \in \mathbb{Z}$, $k_0 \geq 1$ y $\alpha^* \in]0, 1[$, si se cumple que*

$$\alpha(k = k_0, t = 1, \lambda = 2 \mid C_p) > \alpha^*,$$

entonces

$$\alpha(k, t, \lambda \mid C_p) > \alpha^* \quad \forall k, t \in \mathbb{Z}, k \geq k_0, t \geq 1, \lambda \geq 2.$$

Demostración. Si $\alpha(k = k_0, t = 1, \lambda = 2 \mid C_p) > \alpha^*$, entonces por (4.104) (corolario 4.9) sabemos que

$$\alpha(k = k_0, t, \lambda = 2 \mid C_p) > \alpha^*, \text{ para todo } t \in \mathbb{Z}, t \geq 1. \quad (4.136)$$

Aplicando el resultado (4.88) (corolario 4.7) sobre la desigualdad (4.136), obtenemos:

$$\alpha(k, t, \lambda = 2 \mid C_p) > \alpha^*, \text{ para } k, t \in \mathbb{Z}, k \geq k_0, t \geq 1. \quad (4.137)$$

Por último, la proposición 4.2 asegura que α es creciente como función de λ , fijados el resto de parámetros. Por tanto, aplicando dicho resultado a la desigualdad (4.137) tenemos:

$$\alpha(k, t, \lambda \mid C_p) > \alpha^*, \text{ para } k, t \in \mathbb{Z}, k \geq k_0, t \geq 1, \lambda \geq 2,$$

como se quería demostrar. \square

Esta proposición, aplicada al problema OPC (4.5) o a sus versiones *acotada* OPC-b (4.6) o *semiacotada* OPC-sb (4.7), permite identificar zonas de la región factible en las que se puede asegurar que no existirá ninguna solución posible. Efectivamente, en caso de encontrar un valor k_0 para el cual $\alpha(k = k_0, t = 1, \lambda = 2 \mid C_p) > \alpha^*$, entonces, según la proposición que acabamos de probar, no sólo se produce *infactibilidad por α* en todas las soluciones que compartan el mismo valor de k (cosa que ya se había puesto de manifiesto de alguna manera en la proposición 4.8), sino en todas aquellas con $k \geq k_0$. Este resultado constituye, por tanto, la base de

una estrategia para *descartar* zonas de búsqueda que será incorporada al algoritmo que desarrollaremos en el capítulo siguiente.

Es más: del hecho de que la probabilidad α sea creciente como función de k —fijados el resto de parámetros— y tendente asintóticamente al valor 1 (véase el corolario 4.7) puede concluirse que siempre va a existir un valor k_0 suficientemente grande (en principio, sin ningún tipo de relación con $K_{\text{máx}}$) que verifique $\alpha(k = k_0, t = 1, \lambda = 2 | C_p) > \alpha^*$; y, por tanto, por el resultado que acabamos de demostrar, la zona del espacio de búsqueda con k igual o superior a k_0 no proporcionará ninguna solución factible y podrá ser descartada. Lo formalizamos a continuación.

Corolario 4.14. *Dado $\alpha^* \in]0, 1[$, existe $k_0 \in \mathbb{Z}$, $k_{\text{máx}} \geq 1$ tal que*

$$\alpha(k, t, \lambda | C_p) > \alpha^* \quad \forall k, t \in \mathbb{Z}, k \geq k_0, t \geq 1, \lambda \geq 2. \quad (4.138)$$

Demostración. Demostraremos que existe un valor $k_0 \in \mathbb{Z}$, $k_0 \geq 1$ tal que $\alpha(k = k_0, t = 1, \lambda = 2 | C_p) > \alpha^*$, sean cuales sean los valores de los parámetros C_p y α^* , y ya tendremos probado el resultado gracias a la proposición 4.10.

En efecto, por reducción al absurdo, supongamos que *no* existe ningún valor $k_0 \in \mathbb{Z}$, $k_0 \geq 1$ tal que $\alpha(k = k_0, t = 1, \lambda = 2 | C_p) > \alpha^*$. Ello significa que

$$\alpha(k, t = 1, \lambda = 2 | C_p) \leq \alpha^* \quad \forall k \in \mathbb{Z}, k \geq 1. \quad (4.139)$$

Pero entonces, aplicando límites a ambos lados de la expresión (4.139) obtenemos:

$$\lim_{k \rightarrow +\infty} \alpha(k, t = 1, \lambda = 2 | C_p) \leq \alpha^* < 1,$$

lo cual es una contradicción con el hecho probado en el corolario 4.7 de que (4.90) $\lim_{k \rightarrow +\infty} \alpha(k, t = 1, \lambda = 2 | C_p) = 1$.

Ello demuestra que existe un k_0 tal que $\alpha(k = k_0, t = 1, \lambda = 2 | C_p) > \alpha^*$. Por la proposición 4.10, esto implica

$$\alpha(k, t, \lambda | C_p) > \alpha^* \quad \forall k, t \in \mathbb{Z}, k \geq k_0, t \geq 1, \lambda \geq 2,$$

como se quería probar. □

Otro resultado que se deduce de la proposición 4.10 es el que se enuncia y demuestra a continuación.

Corolario 4.15. *Fijados $K_{\text{máx}}, T_{\text{máx}} \in \mathbb{Z}$, $K_{\text{máx}}, T_{\text{máx}} \geq 1$, $\alpha^* \in]0, 1[$, $\beta^* \in]0, 1[$ y $\delta^* > 0$, consideramos los problemas de optimización OPC (4.6) y OPC-sb (4.7):*

$$\left. \begin{array}{l}
 \text{Min} \quad f(k, t, \lambda \mid C_p) \\
 \text{s.a:} \quad \alpha(k, t, \lambda \mid C_p) \leq \alpha^* \\
 \quad \quad \beta(k, t, \lambda \mid C_p, \delta^*) \leq \beta^* \\
 \quad \quad k, t \geq 1, \lambda \geq 2 \\
 \quad \quad k, t \in \mathbb{Z}, \lambda \in \mathbb{R}
 \end{array} \right\} \text{(OPC),}$$

$$\left. \begin{array}{l}
 \text{Min} \quad f(k, t, \lambda \mid C_p) \\
 \text{s.a:} \quad \alpha(k, t, \lambda \mid C_p) \leq \alpha^* \\
 \quad \quad \beta(k, t, \lambda \mid C_p, \delta^*) \leq \beta^* \\
 \quad \quad 1 \leq k \leq K_{\text{máx}} \\
 \quad \quad 1 \leq t \leq T_{\text{máx}} \\
 \quad \quad \lambda \geq 2 \\
 \quad \quad k, t \in \mathbb{Z}, \lambda \in \mathbb{R}
 \end{array} \right\} \text{(OPC-sb),}$$

definidos ambos sobre un proceso con capacidad potencial conocida $C_p > 0$.

Si se verifica

$$\alpha(k = 1, t = 1, \lambda = 2 \mid C_p) > \alpha^*, \tag{4.140}$$

entonces los problemas OPC y OPC-sb no tienen solución (su región factible es vacía).

Demostración. Si suponemos que se cumple (4.140)

$$\alpha(k = 1, t = 1, \lambda = 2 \mid C_p) > \alpha^*,$$

entonces por la proposición 4.10 se tiene:

$$\alpha(k, t, \lambda | C_p) > \alpha^* \quad \forall k, t \in \mathbb{Z}, k \geq 1, t \geq 1, \lambda \geq 2,$$

lo cual significa que absolutamente *todas* las posibles soluciones para (k, t, λ) incumplen la primera restricción de los problemas OPC (4.5) y OPC-sb (4.7). Ello conlleva que ambos problemas son, en tal caso, imposibles, como se quería demostrar. \square

4.5. Extensión a otras distribuciones de probabilidad⁸

Tanto las demostraciones incluidas en las secciones anteriores como los gráficos que acompañan a dichas demostraciones se refieren a los problemas OPC (4.5) y OPC-sb (4.7) y a las funciones involucradas en ellos, en los cuales se supone que la característica de calidad de interés sigue una distribución normal. No obstante, es importante destacar que las expresiones utilizadas para p_Q y f como funciones de p_g y p_y , k y t son totalmente genéricas, en el sentido de que no dependen de la distribución de probabilidad subyacente. Por otro lado, los comportamientos decreciente y creciente, respectivamente, de p_g y p_y demostrados en la proposición 4.1 y el corolario 4.1, los cuales son usados de manera auxiliar en muchas de las demostraciones aludidas, se mantendrían cualquiera que fuese la ley de probabilidad por la que se rigiese la característica de calidad objeto de interés.

Todo ello nos lleva a plantear la hipótesis de que los resultados demostrados en esta sección no dependen, en esencia, de la distribución de probabilidad de la característica de calidad estudiada. En particular,

⁸Si bien es cierto que gran parte de la tesis —incluido este capítulo— está desarrollada bajo el supuesto de normalidad de la variable aleatoria subyacente, consideramos de gran interés dar respuesta aquí, de manera breve, a una pregunta lógica y pertinente desde un punto de vista práctico: ¿continuarían siendo aplicables las propiedades demostradas en el capítulo 4 en el caso de que la característica de calidad siguiese otra distribución distinta de la normal?

continuaría siendo válida la tesis del teorema 4.1, que afirma que la probabilidad de cualificar el proceso desciende según se estrecha la zona verde, fijados el resto de parámetros, así como también el resultado del teorema 4.2, que caracteriza el comportamiento de la función objetivo como función de λ , fijados el resto de parámetros. Y como consecuencia, también serían válidas, independientemente de la distribución de probabilidad subyacente, las ideas principales de resultados como el teorema 4.3, el corolario 4.13 y la proposición 4.10, que sirven de base al método para la resolución del problema OPC que desarrollaremos en el capítulo 5.

Queda fuera del alcance de esta memoria abordar de manera formal y rigurosa la comprobación de dicha hipótesis y, en general, el estudio de las propiedades de las funciones involucradas en el problema del precontrol óptimo para otras distribuciones de probabilidad, de modo similar a como se ha hecho con el caso normal en el presente capítulo.

De todas formas, tras realizar un primer estudio detallado preliminar con la única pretensión de señalar el camino a seguir, parece claro que, para que todo lo demostrado en este capítulo sea aplicable con exactitud⁹, las únicas condiciones que debería cumplir la distribución de probabilidad de la característica de calidad objeto de interés serían las siguientes:

1. Que su función de distribución acumulada no presente discontinuidades¹⁰.
2. Que su función de distribución acumulada sea *estrictamente* creciente al menos dentro del rango de especificaciones.

Si alguna de estas condiciones no se cumple, algunos de los resultados demostrados podrían dejar de ser ciertos. Por ejemplo, si la segunda condición no se verifica, todo lo relacionado con la unicidad de $\lambda_\alpha(k, t)$,

⁹Obviamente, excluimos aquellos cálculos que dependen de la distribución concreta subyacente, como por ejemplo el de la derivada de las probabilidades de pieza verde y amarilla como función de λ (4.11) y (4.17).

¹⁰Equivale a decir que la variable aleatoria subyacente sea continua en sentido estricto, excluyendo el caso de que se trate de una distribución mixta.

$\lambda_\beta(k, t)$ o $\lambda_{\text{máx}}(k, t)$ dejaría de ser válido. O si se estuviera tratando con una distribución discreta, o bien de naturaleza mixta (mezcla de una discreta y una continua), entonces podría darse el caso de que la probabilidad de verde cuando λ tiende a $+\infty$ (es decir, cuando la anchura de la zona verde tiende a cero) no se acercase asintóticamente al valor cero, sino a una constante no nula, lo cual modificaría ligeramente algunos de los resultados probados. Aun así, como se decía arriba, *sí* se mantendrían válidas las *ideas generales* que han guiado las demostraciones.

En especial, nótese que no se ha hecho uso en ninguna parte de este capítulo de la propiedad de simetría de la distribución normal. Por tanto, todos los resultados enunciados son igualmente válidos, en particular, para distribuciones de probabilidad asimétricas, tales como la Weibull, la exponencial, etc., siempre que respeten las condiciones arriba enunciadas.

Así, puede concluirse que las propiedades del problema de OPC deducidas son extensibles, con mayor o menor exactitud, a cualquier distribución de probabilidad que quiera considerarse para la característica de calidad objeto de estudio, incluso de tipo discreto.

4.6. Conclusiones

En este capítulo se ha estudiado desde un punto de vista matemático, formal y teórico el comportamiento de las funciones involucradas en los problemas OPC (4.5), OPC-b (4.6) y OPC-sb (4.7).

Se han enunciado y demostrado propiedades que son observables en un análisis exploratorio, tales como el comportamiento estrictamente decreciente de la probabilidad de cualificar el proceso y la unimodalidad de la función objetivo del OPC, ambas consideradas como funciones del parámetro λ . Asimismo, se han formalizado las consecuencias que estas y otras propiedades tienen en la resolución del problema OPC, en cualquiera de sus tres variantes. Por último, se ha abordado de manera preliminar la extensión de los resultados obtenidos a otras distribuciones de probabilidad distintas de la normal, que es la que ha servido de base para las demostraciones realizadas.

Cabe destacar que todas las propiedades enunciadas y demostradas son válidas para cualquier valor posible de los parámetros implicados ($C_p, \alpha^*, \beta^*, \delta^*, K_{\text{máx}}, T_{\text{máx}}, L_{\text{máx}}$), y no sólo los más usuales.

Como ya se ha resaltado a lo largo del capítulo en varias ocasiones, este estudio teórico hace posible la elaboración de un algoritmo específico exacto para la resolución del problema OPC-sb (4.7), que será desarrollado en el capítulo siguiente.

Un algoritmo exacto para la determinación de planes de precontrol óptimos

5.1. Introducción

En la sección 1.4 hemos expuesto las ventajas de aplicar técnicas de optimización en el diseño de herramientas para el control de la calidad, citando ejemplos significativos en cuanto a técnicas utilizadas y aplicaciones.

En este capítulo abordamos el diseño de un algoritmo exacto específico para la resolución del problema del precontrol óptimo, en el caso en que los límites de precontrol se consideren simétricos respecto al valor nominal y la característica de calidad de interés se suponga normalmente distribuida (problema OPC (4.5)).

Dicho problema ha sido objeto de definición y de un primer estudio en el capítulo 3, donde se ha visto que, dependiendo de las condiciones iniciales (o parámetros del modelo) y de otros aspectos tales como la estrategia empleada, la implementación concreta del modelo, etc., el *software* de optimización de propósito general puede tener dificultades para encontrar el óptimo global. Esa es la razón por la que, tal y como avanzábamos en la sección 3.6, nos planteamos el diseño e implementación de técnicas específicas para tratar de localizar la solución óptima global

del problema OPC (4.5), basadas en el conocimiento que tenemos del mismo, el cual se materializa en el capítulo 4. Ésta (el desarrollo de algoritmos ad hoc) es una alternativa, ya mencionada en la subsección 3.5.1, eficaz y bastante utilizada, válida cuando el uso de *software* genérico no proporciona resultados satisfactorios.

En realidad, el problema que resolveremos es la versión *semiacotada* del problema original, o problema OPC-sb (4.7). La única diferencia entre el problema OPC (4.5) y el problema OPC-sb (4.7) es que, en este último, las variables de decisión k y t están acotadas superiormente por sendos valores determinados por el usuario, $K_{\text{máx}}$ y $T_{\text{máx}}$, respectivamente. Ya se expuso en la subsección 3.5.2 que este tipo de adaptaciones, consistentes en acotar la búsqueda dentro de un determinado recinto, son en cierto modo inevitables si se desea garantizar la convergencia de los algoritmos empleados. También se explicó que, en el problema que nos ocupa, las cotas sobre k y t son hasta cierto punto razonables, pues un plan de precontrol que requiriese observar un elevado número de piezas para poder tomar una decisión no sería aceptable en la práctica. Todo esto está aún más justificado, si cabe, tras haber demostrado analíticamente cuál es el comportamiento del número esperado de piezas a muestrear f (función objetivo de los problemas (4.5) y (4.7)) cuando los parámetros k y t aumentan de valor (véanse las proposiciones 4.3, 4.5 y 4.7).

Por tanto, aquí optamos por *no* acotar superiormente el parámetro λ (a diferencia de lo que se hizo en las pruebas numéricas de la subsección 3.5.2, porque en parte así lo aconsejaba el *software* a utilizar); sabemos que no es estrictamente necesario, al haber demostrado mediante el teorema 4.3 que el rango de valores factibles para λ , fijadas k y t , es un conjunto cerrado y acotado, y por tratarse además f de una función continua respecto de λ en todo su dominio (teorema 4.2).

Tras haber realizado diversas pruebas con diferentes enfoques, presentamos en este capítulo el algoritmo que hemos desarrollado para resolver el problema OPC-sb (4.7), basado en una estrategia enumerativa implícita y en técnicas de búsqueda de raíces sin uso de derivadas.

En la sección siguiente justificamos la elección de tales metodologías

como fundamento de nuestro algoritmo. Posteriormente, en las secciones 5.3 y 5.4 describiremos de manera detallada los algoritmos desarrollados; esto es: un algoritmo auxiliar para la optimización local del parámetro λ basado en métodos numéricos para la búsqueda de raíces sin uso de derivadas, y una búsqueda enumerativa implícita que hace uso del primero. En las secciones posteriores se expondrán los detalles relativos a la implementación de estos algoritmos y a los experimentos computacionales llevados a cabo para su evaluación. Por último, en la sección 5.7 presentaremos las conclusiones que pueden extraerse de dichas pruebas numéricas.

5.2. Estrategias de búsqueda de soluciones

En esta sección exponemos los motivos que justifican la selección de determinados enfoques para el desarrollo de nuestras estrategias de búsqueda.

5.2.1. Mejora local del parámetro λ

Como se ha visto en la sección 4.4, una de las propiedades más interesantes que presenta el problema del OPC (4.5) es la posibilidad de *optimizar* el parámetro λ , fijados los valores de las variables k y t , aplicando simplemente técnicas de búsqueda de raíces de funciones; en concreto, determinando los valores de λ que hacen que se cumplan con igualdad las restricciones respecto de los errores α y β_{δ^*} (véanse (4.118) y (4.125)). A dichos valores los denotamos de manera simplificada como λ_α y λ_β , respectivamente (véase el teorema 4.3).

El algoritmo que presentaremos en las secciones siguientes hará uso de esta propiedad. Dado que, en general, no resulta muy costoso computacionalmente calcular las raíces de una determinada función, se implementará una estrategia de *mejora* del parámetro λ que se aplicará de manera sistemática a toda solución candidata (k, t, λ) (a no ser que, por ejemplo, previamente sea descartada por algún motivo). De ese modo,

la solución candidata será sustituida por la solución mejorada (k, t, λ') (donde λ' se calcula como el mejor entre λ_α y λ_β , según la expresión (4.131)). Este procedimiento puede ser considerado una estrategia de *búsqueda local* (Michalewicz y Fogel, 2000).

En cierto sentido, al aplicar esta estrategia estamos *fijando* el valor de la variable λ en función del valor que tomen las otras dos variables, k y t , disminuyendo así la *dimensión* del espacio de soluciones y reduciendo la búsqueda a puntos del estilo (k, t) , tomando k y t valores enteros positivos.

La técnica de mejora local de λ , que desarrollaremos en la sección 5.3, será aplicada de manera auxiliar y conjunta con la estrategia de búsqueda principal que presentaremos más adelante.

Dado el papel destacado que desempeña la búsqueda de raíces de funciones en el desarrollo de esta técnica de mejora, realizamos a continuación un resumen de los métodos relacionados y analizamos cuáles pueden ser los más adecuados en nuestro caso concreto.

Métodos de búsqueda de raíces

De manera genérica, se conocen como métodos de búsqueda de raíces de funciones (*root finding methods*, en inglés) aquellos procedimientos numéricos —iterativos— orientados a resolver ecuaciones de la forma $F(x) = 0$, donde F puede representar una función real de variable real, o bien una función de \mathbb{R}^m en \mathbb{R}^m (en tal caso, se tendría en realidad un sistema de m ecuaciones con m incógnitas a despejar). Como es sabido, en ambos contextos, si la ecuación anterior se verifica para un determinado valor x_0 de x se dice que x_0 es *raíz* de la función F .

Los métodos de búsqueda de raíces para el caso unidimensional (que es el que nos interesa aquí) pueden clasificarse en dos grandes grupos: aquellos que *no* hacen uso de la derivada de la función, y aquellos que sí la utilizan. La mayoría de los primeros requieren como punto de partida un intervalo $[a, b]$ dentro del cual haya la seguridad de encontrar una raíz (en concreto, se suele exigir la condición $F(a)F(b) < 0$), razón por

la cual son etiquetados con el término inglés *bracketing methods*¹, que podría traducirse en español como *métodos de acotación*. Los algoritmos de búsqueda de raíces que no necesitan acotarlas mediante intervalos son conocidos como *métodos abiertos* u *open methods* (Chapra y Canale, 2010). En todos los casos, uno o varios valores *candidatos a raíz* son necesarios para iniciar estos algoritmos, en tanto en cuanto se trata de procedimientos iterativos. Los métodos de acotación suelen ser menos rápidos que los abiertos, pero convergen siempre, bajo ciertas condiciones básicas, independientemente de los valores de partida. En los métodos abiertos la convergencia depende, en parte, de la cercanía a la raíz del punto o puntos iniciales elegidos.

Entre los algoritmos sin uso de derivadas más clásicos pueden destacarse el método de *bisección*, el de la *secante* y el de *regula falsi* o *falsa posición*. Todos ellos basan su funcionamiento en resultados o mecanismos sencillos y conocidos (como el teorema del valor medio, en el caso del método de bisección, o la realización de sucesivas interpolaciones lineales en el de *regula falsi*), y su convergencia es más bien lenta (lineal en el caso del algoritmo de bisección, y del orden de $\varphi = 1,618\dots$ —número áureo— en el de la secante). El de la secante se trata de un método abierto y por tanto, como se ha dicho, puede no converger en determinadas circunstancias. Véanse, por ejemplo, Press et al. (2007) y Burden y Faires (2010) para más referencias sobre estos métodos.

Otros algoritmos sin uso de derivadas (y de acotación; por tanto, con convergencia prácticamente asegurada) más sofisticados y rápidos que merece la pena reseñar son el *algoritmo de Ridders* (Ridders, 1979) y el de *Brent* (Brent, 1971, 1973). El algoritmo de Ridders puede ser visto como una mejora del método de *regula falsi*, en la cual en lugar de realizar interpolaciones lineales se utiliza una interpolación exponencial.

¹También se conocen con este nombre los procedimientos cuyo objetivo es precisamente encontrar un intervalo que contenga alguna raíz. La terminología es ambigua, pues también la expresión *búsqueda incremental* se usa para representar ambos conceptos. En esta memoria entenderemos que un método de búsqueda incremental sirve para encontrar un intervalo inicial que contenga una raíz y que servirá de punto de partida a un *bracketing method* o método de acotación.

Presenta convergencia cuadrática, pero como en cada iteración se realizan *dos* evaluaciones de la función F , el orden real de convergencia es $\sqrt{2} = 1,414\dots$

El algoritmo de Brent es considerado habitualmente el método de elección cuando no se dispone de la derivada de la función F o no se desea usarla (Press et al., 2007). Combina el método de bisección y el de la secante con una interpolación cuadrática inversa y su convergencia es supralineal (concretamente presenta una tasa de convergencia entre 1, que es la que corresponde al método de bisección, y 1,839, que es la de la interpolación cuadrática inversa, de acuerdo con Epperson, 2007); realiza una llamada a F en cada iteración.

En cuanto a los métodos unidimensionales de búsqueda de raíces con uso de derivadas, la referencia más conocida es el *método de Newton-Raphson* (véase, por ejemplo, Ypma, 1995), que presenta una rápida convergencia —cuadrática— si el punto inicial se encuentra próximo a la raíz buscada, pero que, como es sabido, puede diverger o incluso entrar en un ciclo no convergente si el punto de partida no es el adecuado. Esta dificultad puede ser salvada mediante modificaciones oportunas, como por ejemplo la hibridación con el método de bisección propuesto en Press et al. (2007).

Más información sobre los métodos de búsqueda de raíces puede encontrarse en Mathews y Fink (2004), Press et al. (2007) y Chapra y Canale (2010). En Soleymani y Sharifi (2009) se analizan algunas propuestas más recientes en el ámbito de métodos sin uso de derivadas.

Cabe resaltar que estos métodos suelen ocuparse sólo de encontrar una raíz, y en general no ofrecen ninguna información acerca de la existencia o no de más raíces.

La relativa complejidad de las expresiones de las funciones involucradas en nuestro problema y, sobre todo, la presencia en ellas de la función de distribución normal tipificada (la cual, entre otras cosas, se ha mostrado cómo limitó la selección de algoritmos para la prueba numérica de la subsección 3.5.2) hacen aconsejable la elección de un método de búsqueda de raíces sin uso de derivadas. Por otro lado, vistas la continuidad, la

monotonía y otras propiedades de α y β como funciones de λ , fijadas k y t (véanse los corolarios 4.2 y 4.3), es de esperar que cualquier método sencillo de búsqueda de raíces funcione razonablemente bien cuando se aplique a las mismas.

Métodos de búsqueda de raíces versus métodos de optimización

El estudio de las funciones que componen las restricciones del problema (4.5) nos ha conducido a plantear un método de optimización local del parámetro λ que se apoya en la búsqueda de raíces, cuyo objetivo es dar solución al problema de optimización no lineal OPC-NLP (4.117) introducido en el capítulo 4. Otro enfoque posible, más genérico en el sentido de que no habría requerido un análisis tan profundo de las propiedades de dichas funciones, habría sido aplicar métodos de optimización de funciones reales de variable real a ese problema.

Cabe decir que, en términos computacionales, puede reconocerse cierta equivalencia entre los métodos de búsqueda de raíces y los de optimización de funciones, ya que algunos de estos últimos, en definitiva, terminan buscando puntos en los que la derivada se iguala a cero.

La diferencia está, obviamente, en la función o funciones que *concentran* el esfuerzo computacional: mientras que normalmente cualquier método numérico de optimización va a tener que evaluar repetidamente la función objetivo y/o su derivada, el método que estamos proponiendo requiere que las funciones que forman parte de las restricciones, α y β , sean evaluadas varias veces en cada ocasión que queramos optimizar el valor de λ (debido al funcionamiento iterativo de los métodos numéricos de búsqueda de raíces); en cambio, tal y como se ha descrito, la función objetivo f sólo será evaluada en los puntos *candidatos* λ_α y λ_β , una vez éstos sean hallados.

Dado que el problema OPC-NLP (4.117) contiene restricciones, estaríamos refiriéndonos a aplicar métodos de *optimización restringida*, como por ejemplo el método de los *multiplicadores de Lagrange* (véanse, por ejemplo, Harris, 2007; Thomas Jr. et al., 2014) o las estrategias que hacen uso de una *función de penalización* (*penalty function*, en inglés; Carroll, 1961;

Fiacco y McCormick, 1966). En ambos enfoques, el problema original con restricciones se transforma en uno irrestringido equivalente; los lados izquierdos de las restricciones (en nuestro caso, las funciones α y β) y la función objetivo se unen para componer una única función, que es la que se evalúa (ella y/o su derivada) en cada iteración del método de optimización irrestringida que se decida aplicar. Son aplicables métodos propios de optimización irrestringida como, por ejemplo, los desarrollados por León et al. (1998) y Sanmatías y Vercher (1998), entre otros muchos. Sendas revisiones bastante completas sobre los métodos de función de penalización y de lagrangiano y sobre estrategias para manejo de restricciones en el ámbito de optimización metaheurística pueden encontrarse en Boukari y Fiacco (1995) y Coello Coello (2002), respectivamente.

En consecuencia, puede afirmarse que con nuestra propuesta se consigue incluso cierto ahorro computacional, ya que usando un método *penalty function* estándar o el método de los multiplicadores de Lagrange no evitaríamos tener que evaluar α y β y/o sus derivadas tantas veces como la propia función objetivo f .

5.2.2. Enumeración implícita de las soluciones

Tal y como se acaba de exponer, la aplicación sistemática del método de optimización o mejora local de λ hace que la búsqueda de la mejor solución para el problema OPC-sb (4.7) pueda reducirse a un *barrido* de todas las soluciones de tipo $(k, t, \lambda'(k, t))$ donde $k, t \in \mathbb{Z}$, $1 \leq k \leq K_{\text{máx}}$, $1 \leq t \leq T_{\text{máx}}$, siendo $K_{\text{máx}}$ y $T_{\text{máx}}$ sendos límites superiores convenientemente elegidos. En cierto modo, este planteamiento puede entenderse como una *búsqueda exhaustiva* o *enumerativa* (Michalewicz y Fogel, 2000).

Además, los resultados encontrados en el capítulo 4 (en concreto, la proposición 4.10) permiten también desarrollar reglas *exactas* para *descartar* zonas de búsqueda sin necesidad de evaluarlas, a medida que vaya explorándose el espacio de soluciones, lo que, combinado con lo anterior, daría lugar a lo que se conoce como una estrategia *enumerativa*

implícita o abreviada.

Teniendo en cuenta todo lo dicho, y siempre suponiendo que el procedimiento auxiliar de mejora local del parámetro λ no resulte excesivamente costoso, tal estrategia de búsqueda enumerativa implícita parece un enfoque adecuado y sencillo para resolver de manera *exacta* el problema (4.7). Esta será la base, por tanto, del algoritmo que desarrollaremos en la sección 5.4.

5.3. Algoritmo de mejora local de λ

A continuación se describe la función `OPC_improve_lambda`, un procedimiento de *mejora local* del parámetro λ cuyos fundamentos hemos introducido en el apartado 5.2.1. Como ya se ha dicho en la sección anterior, `OPC_improve_lambda` será utilizado de manera auxiliar en el algoritmo que se han desarrollado para el problema OPC-sb (4.7).

Recordemos que, fijadas k y t , la idea es buscar $\lambda'(k, t)$, el valor de λ que, proporcionando una solución factible al problema OPC-NLP(k, t) (4.117), minimice la expresión $f(k, t, \lambda)$. Como se estudió en la sección 4.4, dicha búsqueda se reduce a determinar los valores $\lambda_\alpha(k, t)$ y $\lambda_\beta(k, t)$ que hacen que se verifiquen con igualdad las restricciones (4.118) y (4.125), respectivamente. Más concretamente, el teorema 4.3 recoge todos los resultados relativos a esta característica del problema que nos ocupa, y que se materializan en el algoritmo 5.1.

El algoritmo requiere hacer uso de un método de búsqueda de raíces, como ya se ha mencionado anteriormente. La técnica elegida ha sido el método de Brent (Brent, 1971, 1973; véase la subsección 5.2.1). Asimismo, proponemos una rutina sencilla de búsqueda incremental (Chapra y Canale, 2010) para, en caso de ser necesario, determinar un intervalo que contenga una raíz y que pueda ser usado como punto de partida del algoritmo, en cada llamada a éste².

²Los métodos de búsqueda incremental localizan un cambio de signo en la función objeto de estudio a base de evaluar su valor en un subconjunto discreto de puntos de su dominio; la idea es ir dando *saltos*, de longitud constante o variable, a lo largo del

Algoritmo 5.1: Función OPC_improve_lambda(sol₀)

Entrada: Solución a mejorar sol₀ = (k₀, t₀, λ₀).

Salida: Solución mejorada sol'₀ = (k₀, t₀, λ'₀).

// Usar un método adecuado de búsqueda de raíces para obtener λ_α y λ_β:

λ_α = λ tal que α(k₀, t₀, λ) = α*

si S(C_p, δ*)^{k₀} < β* **entonces**

λ_β = 2

en otro caso

λ_β = λ tal que β(k₀, t₀, λ, δ*) = β*

// Según lo visto en el teorema 4.3:

si f(k₀, t₀, λ_α) < f(k₀, t₀, λ_β) **entonces**

λ'₀ = λ_α // λ_α proporciona el mínimo valor de f, fijados k₀ y t₀

en otro caso

λ'₀ = λ_β // λ_β proporciona el mínimo valor de f, fijados k₀ y t₀

sol'₀ = (k₀, t₀, λ'₀)

// El mismo teorema proporciona una manera rápida de comprobar si la solución mejorada es factible:

si λ_β ≤ λ_α **entonces**

sol'₀.es_factible = verdadero

en otro caso

sol'₀.es_factible = falso

// Guardamos la información de λ_α y λ_β:

sol'₀.λ_α = λ_α

sol'₀.λ_β = λ_β

devolver sol'₀

dominio e ir comprobando si en algún momento se produce un cambio de signo (todo ello, suponiendo que la función no presente discontinuidades). Como es lógico, estos métodos pueden tener dificultades para detectar raíces relativamente cercanas entre sí (porque si uno de esos saltos abarca un número par de raíces, no se detectará ningún

Como puede observarse en el algoritmo 5.1, la función `OPC_improve_lambda` no verifica la existencia del parámetro λ_α para los valores de k y t considerados (véase el teorema 4.3). Esto será tenido en cuenta en el algoritmo que hará uso de dicha función, mediante un *filtro* previo a su llamada.

5.3.1. Ejemplos de aplicación

A continuación mostramos cómo se lleva a cabo la determinación de $\lambda'(k, t)$ con la función `OPC_improve_lambda`, a través de dos ejemplos sencillos: un caso factible y uno infactible por $\lambda_\alpha < \lambda_\beta$ (véanse las explicaciones posteriores a la demostración del teorema 4.3). El caso de infactibilidad por α no puede darse en este algoritmo porque es filtrado previamente, como ya se ha explicado.

Ejemplo 1: Solución factible

Sean los parámetros del modelo (4.117) $C_p = 1,2$, $\alpha^* = 0,01$, $\beta^* = 0,1$ y $\delta^* = 1,5462$, de modo que $p^* = 2\%$ (véase (2.34)), lo cual se corresponde con el ejemplo 9 de la batería de problemas de la experiencia numérica de la subsección 3.5.2. Supongamos que deseamos *optimizar el valor de λ* para $k_0 = 5$ y $t_0 = 5$. Siguiendo los pasos del algoritmo 5.1, tendríamos:

Paso 1 Determinar λ_α tal que $\alpha(k_0, t_0, \lambda_\alpha) = \alpha^*$. Aplicando el método de Brent se obtienen las siguientes aproximaciones:

- Iteración 1.^a: $\lambda_\alpha = 4,74688472898800$; residuo = $-0,00730992$.
- Iteración 2.^a: $\lambda_\alpha = 5,70438120499305$; residuo = $-0,00352436$.
- Iteración 3.^a: $\lambda_\alpha = 5,97276044969235$; residuo = $-0,000897272$.
- Iteración 4.^a: $\lambda_\alpha = 6,05022306865086$; residuo = $7,51840 \cdot 10^{-5}$.
- Iteración 5.^a: $\lambda_\alpha = 6,04423416666998$; residuo = $-3,88148 \cdot 10^{-6}$.

cambio de signo en la función). En nuestro caso *no* existe este problema, ya que α es estrictamente creciente y β estrictamente decreciente como funciones de λ , fijados el resto de parámetros (corolarios 4.2 y 4.3), por lo que no habrá multiplicidad de raíces.

- Iteración 6.^a: $\lambda_\alpha = 6,04452817410819$; residuo = $-1,55824 \cdot 10^{-8}$.
- Iteración 7.^a: $\lambda_\alpha = 6,04452935892911$; residuo = $1,99927 \cdot 10^{-14}$.
- Iteración 8.^a: $\lambda_\alpha = 6,04452935892759$; residuo = $-7,68482 \cdot 10^{-16}$.

Por tanto, $\lambda_\alpha = 6,04452935892759$.

Paso 2 Comprobar si $S(C_p, \delta^*)^{k_0} < \beta^*$:

- $S(C_p, \delta^*)^{k_0} = 0,98^5 = 0,903921$.
- $\beta^* = 0,1$.

Como $0,903921 \not< 0,1$, vamos al **Paso 3**.

Paso 3 Determinar λ_β tal que $\beta(k_0, t_0, \lambda_\beta) = \beta^*$. Aplicando el método de Brent se obtienen las siguientes aproximaciones:

- Iteración 1.^a: $\lambda_\beta = 6,05757565710014$; residuo = $-0,0401084$.
- Iteración 2.^a: $\lambda_\beta = 5,40792437335796$; residuo = $0,0318609$.
- Iteración 3.^a: $\lambda_\beta = 5,69552616202557$; residuo = $-0,00747170$.
- Iteración 4.^a: $\lambda_\beta = 5,64089277916892$; residuo = $-0,00106866$.
- Iteración 5.^a: $\lambda_\beta = 5,63207042465795$; residuo = $8,37533 \cdot 10^{-6}$.
- Iteración 6.^a: $\lambda_\beta = 5,63213902963304$; residuo = $-4,73520 \cdot 10^{-8}$.
- Iteración 7.^a: $\lambda_\beta = 5,63213864393828$; residuo = $-2,08185 \cdot 10^{-12}$.
- Iteración 8.^a: $\lambda_\beta = 5,63213864392132$; residuo = $5,55111 \cdot 10^{-17}$.

Por tanto, $\lambda_\beta = 5,63213864392132$.

Paso 4 Comprobar si $f(k_0, t_0, \lambda_\alpha) < f(k_0, t_0, \lambda_\beta)$:

- $f(k_0, t_0, \lambda_\alpha) = 11,789697624275927$.
- $f(k_0, t_0, \lambda_\beta) = 10,246981850684488$.

Como $11,789697624275927 \not< 10,246981850684488$, vamos al **Paso 5**.

Paso 5 Asignar $\lambda'_0 = \lambda_\beta$:

$$\lambda'_0 = 5,63213864392132.$$

Paso 6 Asignar la solución a devolver $\text{sol}'_0 = (k_0, t_0, \lambda'_0)$:

$$\text{sol}'_0 = (5; 5; 5,63213864392132)$$

Paso 7 Comprobar si $\lambda_\beta \leq \lambda_\alpha$:

- $\lambda_\alpha = 6,04452935892759.$
- $\lambda_\beta = 5,63213864392132.$

Como $5,63213864392132 \leq 6,04452935892759$, vamos al **Paso 8**.

Paso 8 Asignar $\text{sol}'_0.\text{es_factible} = \text{verdadero}.$

Paso 9 Devolver $\text{sol}'_0.$

La figura 5.1 muestra las soluciones que va generando el método de Brent para determinar λ_α y λ_β .

Ejemplo 2: Solución no factible por $\lambda_\alpha < \lambda_\beta$

Consideramos los mismos parámetros para el modelo (4.117): $C_p = 1,2$, $\alpha^* = 0,01$, $\beta^* = 0,1$ y $\delta^* = 1,5462$, de modo que $p^* = 2\%$ (véase (2.34)). Ahora queremos *optimizar el valor de λ* para $k_0 = 5$ y $t_0 = 2$. Seguimos para ello los pasos del algoritmo 5.1:

Paso 1 Determinar λ_α tal que $\alpha(k_0, t_0, \lambda_\alpha) = \alpha^*$. Aplicando el método de Brent se obtienen las siguientes aproximaciones:

- Iteración 1.^a: $\lambda_\alpha = 3,20019056123592$; residuo = $-0,00524182$.
- Iteración 2.^a: $\lambda_\alpha = 3,45864087963014$; residuo = $-0,000868720$.
- Iteración 3.^a: $\lambda_\alpha = 3,49123893798814$; residuo = $-9,52600 \cdot 10^{-5}$.
- Iteración 4.^a: $\lambda_\alpha = 3,49512012512382$; residuo = $6,42508 \cdot 10^{-7}$.
- Iteración 5.^a: $\lambda_\alpha = 3,49509412273372$; residuo = $-2,74406 \cdot 10^{-9}$.

- Iteración 6.^a: $\lambda_\alpha = 3,49509423331405$; residuo = $-7,85951 \cdot 10^{-14}$.
- Iteración 7.^a: $\lambda_\alpha = 3,49509423331722$; residuo = $2,30718 \cdot 10^{-16}$.

Por tanto, $\lambda_\alpha = 3,49509423331722$.

Paso 2 Comprobar si $S(C_p, \delta^*)^{k_0} < \beta^*$:

- $S(C_p, \delta^*)^{k_0} = 0,98^5 = 0,903921$.
- $\beta^* = 0,1$.

Como $0,903921 \not\leq 0,1$, vamos al **Paso 3**.

Paso 3 Determinar λ_β tal que $\beta(k_0, t_0, \lambda_\beta) = \beta^*$. Aplicando el método de Brent se obtienen las siguientes aproximaciones:

- Iteración 1.^a: $\lambda_\beta = 5,25889669522635$; residuo = $-0,0629362$.
- Iteración 2.^a: $\lambda_\beta = 4,42698194197746$; residuo = $0,0151733$.
- Iteración 3.^a: $\lambda_\beta = 4,58858653602846$; residuo = $-0,00845335$.
- Iteración 4.^a: $\lambda_\beta = 4,53076610885424$; residuo = $-0,000652357$.
- Iteración 5.^a: $\lambda_\beta = 4,52613020125974$; residuo = $3,09492 \cdot 10^{-6}$.
- Iteración 6.^a: $\lambda_\beta = 4,52615209111075$; residuo = $-1,07791 \cdot 10^{-8}$.
- Iteración 7.^a: $\lambda_\beta = 4,52615201513662$; residuo = $-1,77247 \cdot 10^{-13}$.
- Iteración 8.^a: $\lambda_\beta = 4,52615201513537$; residuo = $-1,38778 \cdot 10^{-17}$.

Por tanto, $\lambda_\beta = 4,52615201513537$.

Paso 4 Comprobar si $f(k_0, t_0, \lambda_\alpha) < f(k_0, t_0, \lambda_\beta)$:

- $f(k_0, t_0, \lambda_\alpha) = 5,593361393476368$.
- $f(k_0, t_0, \lambda_\beta) = 6,678593621997701$.

Como $5,593361393476368 < 6,678593621997701$, vamos al **Paso 5'**.

Paso 5' Asignar $\lambda'_0 = \lambda_\alpha$:

$$\lambda'_0 = 3,49509423331722.$$

Paso 6 Asignar la solución a devolver $\text{sol}'_0 = (k_0, t_0, \lambda'_0)$:

$$\text{sol}'_0 = (5; 2; 3,49509423331722)$$

Paso 7 Comprobar si $\lambda_\beta \leq \lambda_\alpha$:

- $\lambda_\alpha = 3,49509423331722$.
- $\lambda_\beta = 4,52615201513537$.

Como $4,52615201513537 \not\leq 3,49509423331722$, vamos al **Paso 8'**.

Paso 8' Asignar $\text{sol}'_0.\text{es_factible} = \text{falso}$.

Paso 9 Devolver sol'_0 .

En la figura 5.2 se muestran las iteraciones que ha seguido el método de Brent para determinar λ_α y λ_β en este segundo ejemplo.

5.4. Algoritmo de enumeración implícita para el problema del precontrol óptimo

En esta sección presentamos y describimos con detalle el algoritmo que hemos desarrollado para el problema OPC-sb (4.7) basándonos en una metodología de búsqueda enumerativa implícita.

Como se ha dicho en la subsección 5.2.2, esta búsqueda enumerativa se apoya directamente en la aplicación del procedimiento de *mejora local* del parámetro λ propuesto en la sección 5.3 y concretado en el algoritmo 5.1. Así, la búsqueda consistiría básicamente en ir variando los valores de k y t desde 1 hasta $K_{\text{máx}}$ y $T_{\text{máx}}$, respectivamente, evaluando en cada caso la solución mejorada $(k, t, \lambda'(k, t))$. Los parámetros $K_{\text{máx}}$ y $T_{\text{máx}}$ representan sendas cotas superiores para k y t , elegidas ad hoc por el usuario, o determinadas de manera analítica, usando las propiedades conocidas del problema (concretamente, las relativas al comportamiento asintótico de las funciones involucradas cuando el valor de k o t crece; véanse en particular las proposiciones 4.3, 4.5 y 4.7 y el corolario 4.14).

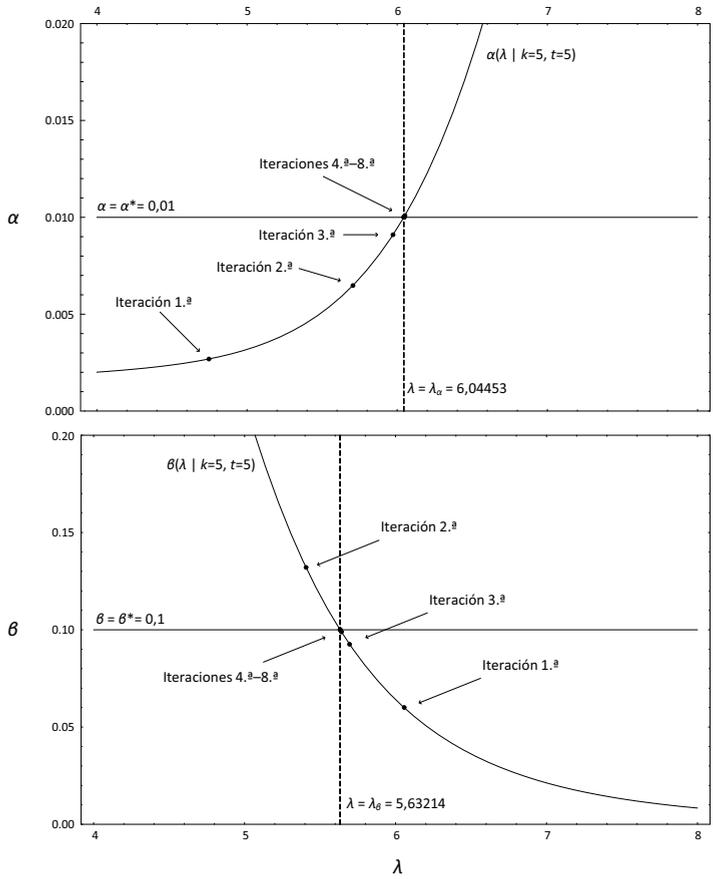


Figura 5.1: Ejemplo de determinación de λ_α y λ_β mediante el método de Brent, para el caso $k_0 = 5$, $t_0 = 5$, con $C_p = 1,2$, $\alpha^* = 0,01$, $\beta^* = 0,1$ y $\delta^* = 1,5462$.

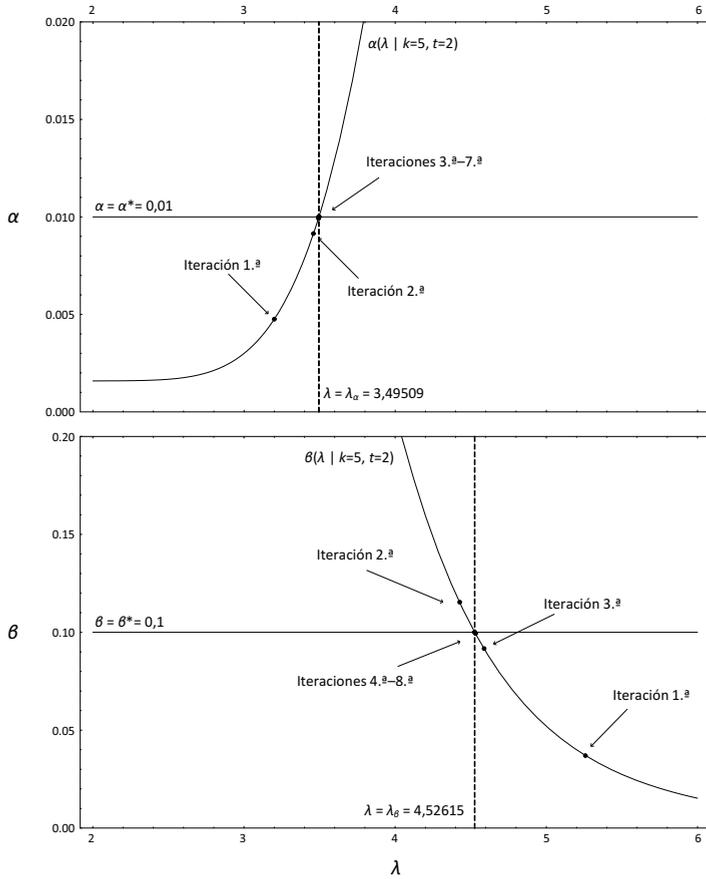


Figura 5.2: Ejemplo de determinación de λ_α y λ_β mediante el método de Brent, para el caso $k_0 = 5$, $t_0 = 2$, con $C_p = 1,2$, $\alpha^* = 0,01$, $\beta^* = 0,1$ y $\delta^* = 1,5462$.

Dentro de este esquema general, pueden plantarse diferentes maneras de *organizar* la búsqueda. Por ejemplo, podríamos optar por un enfoque aleatorio, *visitando* completamente al azar todas las soluciones de la forma $(k, t, \lambda'(k, t))$ hasta completar el dominio de búsqueda. Sin embargo, el estudio realizado en el capítulo anterior sobre el comportamiento de las funciones involucradas en el problema (4.7) nos permite tomar ciertas decisiones en el diseño de este esquema de búsqueda, orientadas a conseguir un procedimiento más eficiente, las cuales detallamos a continuación.

5.4.1. Parámetros del problema y del algoritmo

Recordamos a continuación los parámetros que manejará el algoritmo. En primer lugar, los parámetros propios del problema a resolver (4.7):

- Parámetro del proceso: Capacidad potencial C_p .
- Parámetros del problema: Cotas máximas α^* y β^* , respectivamente para la tasa de falsa alarma y para la probabilidad de no detectar una desviación estandarizada δ^* .
- Parámetros adicionales del problema: Cotas máximas $K_{\text{máx}}$ y $T_{\text{máx}}$ para las variables k y t , respectivamente.

En cuanto al algoritmo, no creemos necesario en este caso establecer ningún criterio de parada clásico, tal como un cota superior para el número de iteraciones a realizar, dado que éstas están ya limitadas de antemano a un máximo de $K_{\text{máx}} \cdot T_{\text{máx}}$ por la misma mecánica a seguir. En todo caso, a la hora de implementar el algoritmo, si se desea, puede añadirse la posibilidad de que el usuario interactúe abortando el procedimiento cuando considere excesivo el tiempo de ejecución. Por tanto, no se incluye ningún parámetro asociado propiamente al algoritmo.

5.4.2. Orden de visita de las soluciones

Diseñamos este procedimiento para que comience explorando el caso $(k = 1, t = 1)$, incrementando después secuencialmente los valores de k y t . Lo hacemos así porque, en el caso de encontrar soluciones factibles con valores pequeños de k y t , es bastante probable que éstas sean *muy buenas*, ya que la función objetivo depende en gran parte de estos dos parámetros: fijados el resto de parámetros (incluida la variable λ), cuantas menos piezas verdes o amarillas consecutivas sea necesario muestrear para finalizar el proceso, menor será el valor esperado del número total de piezas a muestrear, es decir, menor será el valor de la función objetivo (proposiciones 4.3 y 4.5).

Concretamente, tras visitar una solución cualquiera $(k = k_0, t = t_0)$, la siguiente solución a explorar será $(k = k_0 + 1, t = t_0)$, siempre que no hayamos llegado ya al máximo valor de k que se desee explorar; en tal caso, pasaremos a aumentar el valor de t en una unidad y *reiniciaremos* el valor de k , de tal modo que la siguiente solución a evaluar sería $(k = 1, t = t_0 + 1)$. Y así continuaremos hasta que no queden puntos pendientes de explorar.

Cabe hacer notar que, entre otras cosas, el comportamiento de $f(k, t, \lambda'(k, t))$ fijado t *no* tiene por qué ser necesariamente monótono creciente según aumenta el valor de k (véase, de hecho, el ejemplo numérico de la subsección 5.4.5 más adelante).

Nótese por último que, como se ha señalado indirectamente, la búsqueda puede generar soluciones *no factibles* o, dicho de otro modo, puede pasar por puntos (k, t) para los que $\lambda_\beta > \lambda_\alpha$ (véase el teorema 4.3).

Esta manera específica de recorrer las soluciones posibles del problema, junto con el conocimiento que se tiene del mismo, nos permitirá definir una regla para *descartar* zonas de búsqueda, como pasamos a detallar a continuación.

5.4.3. Reducción de la zona de búsqueda

Denotaremos por $k_{\text{máx}}$ al máximo valor de k que es *necesario* explorar. A diferencia de $K_{\text{máx}}$, que representa el máximo valor de k a tener en cuenta, este umbral $k_{\text{máx}}$ es susceptible de ir siendo ajustado —concretamente, reducido— a medida que avanzan las iteraciones del algoritmo. Lógicamente, se verificará siempre la desigualdad $k_{\text{máx}} \leq K_{\text{máx}}$, e inicialmente fijaremos $k_{\text{máx}} = K_{\text{máx}}$. De manera equivalente, también puede considerarse $K_{\text{máx}}$ sencillamente como un valor inicial proporcionado por el usuario para el parámetro $k_{\text{máx}}$.

Como se acaba de mencionar, existen situaciones durante la ejecución del algoritmo en que, por el orden en que se van visitando las soluciones y gracias a las propiedades del problema que estamos resolviendo, el valor de $k_{\text{máx}}$ puede reducirse. En particular, nos referimos al caso de *infactibilidad por α* , que ya se ha mencionado anteriormente, en el que podemos realizar dicha operación con la seguridad de no estar *descartando* soluciones mejores que las ya halladas.

Concretamente, en el caso en que estemos explorando una solución (k_0, t_0, λ_0) tal que $\alpha(k = k_0, t = t_0, \lambda = 2) > \alpha^*$, sabemos por la expresión (4.104) (corolario 4.9) y la proposición 4.10 que no hay ninguna solución factible para el problema (4.7) con $k = k_0$ —independientemente del valor que tomen t y λ — porque se incumplirá la restricción sobre α (3.26), y lo mismo sucede para cualquier solución con $k > k_0$. En nuestro algoritmo realizaremos esta validación antes de intentar la mejora sobre λ de una solución dada, de modo que, en el caso de que se verifique esta *condición de infactibilidad*, el valor de $k_{\text{máx}}$ se actualizará a $k_0 - 1$. Más precisamente, para ahorrar cálculos y dado que la condición no depende del valor de t , la comprobación se realizará únicamente la primera vez que el algoritmo pase por cada valor de k , es decir, cuando $t = 1$ (que, de hecho, es lo que plantea la proposición 4.10).

El efecto práctico del parámetro $k_{\text{máx}}$ en el algoritmo es claro: si se está evaluando una solución $(k = k_0, t = t_0)$ tal que $k_0 > k_{\text{máx}}$, dicha solución puede descartarse y, siguiendo el orden de recorrido de las soluciones visto en la subsección 5.4.2, la siguiente solución a visitar será

($k = 1, t = t_0 + 1$), no evaluándose por tanto el resto de soluciones con $t = t_0$ y $k \geq k_0, k \leq K_{\text{máx}}$.

Nótese que, tal y como está descrita, la reducción del espacio de búsqueda mediante el ajuste del parámetro $k_{\text{máx}}$ se produce durante el recorrido del algoritmo por soluciones con $t = 1$, *independientemente de que éstas sean o no factibles*.

5.4.4. Descripción del algoritmo

Tras presentar los elementos principales que conforman este procedimiento de búsqueda enumerativa, a continuación los unimos en lo que será la forma final del algoritmo, que hemos llamado OPCenum.

Con el fin de ganar en claridad, descomponemos el bucle principal del algoritmo en las siguientes fases o subprocedimientos, que detallamos por separado³.

Procedimiento OPCenum_improve (algoritmo 5.2). Actúa de *enlace* con la función de mejora local de λ OPC_improve_lambda (véase el algoritmo 5.1), realizando un filtro previo para detectar situaciones de infactibilidad mencionadas en el apartado 5.4.3.

Procedimiento OPCenum_update_status (algoritmo 5.3). Realiza las acciones de actualizar el valor del parámetro $k_{\text{máx}}$ y de la mejor solución encontrada.

Procedimiento OPCenum_set_next_step (algoritmo 5.4). Determina la siguiente solución a explorar.

Estos procedimientos se ejecutan de forma secuencial dentro de cada iteración del algoritmo OPCenum, cuya forma final se recoge en el algoritmo 5.5.

³En todos los procedimientos o funciones en pseudocódigo que se presentan en el actual capítulo, las variables que se utilizan tienen alcance *global* por defecto, esto es, a partir del momento en que se definan están disponibles y accesibles por todos los procedimientos o funciones que sean invocados con posterioridad. En particular, utilizaremos *variables globales* para representar parámetros del problema o de los algoritmos correspondientes.

La estructura de un algoritmo enumerativo como el que se presenta aquí es relativamente sencilla. La razón principal por la cual se ha preferido subdividir el algoritmo 5.5 en varios subprocedimientos, además de ganar en claridad como ya se ha dicho, es facilitar la reutilización de esta estructura en el caso de que decidamos abordar en el futuro otros casos más complejos que requieran diseñar algoritmos más elaborados (véase la sección 7.2.9 en el capítulo de trabajos futuros).

Algoritmo 5.2: Procedimiento OPCenum_improve(sol_act)

Entrada: Solución actual a mejorar $\text{sol_act} = (k_{\text{act}}, t_{\text{act}}, \lambda_{\text{act}})$.

Salida: Comprueba si hay infactibilidad para $k = k_{\text{act}}$ debida a la restricción sobre α , y si no la hay sobrescribe sol_act con la solución mejorada $\text{sol_act}'$.

// Comprobamos si para la solución actual va a existir o no λ_α ; por la proposición 4.10, sabemos que esto depende sólo del valor de k . Para no repetir operaciones innecesarias, validaremos si ya hemos pasado antes por ese k ; en nuestro caso, por la manera como se recorren las soluciones, sólo es necesario comprobarlo cuando $t_{\text{act}} = 1$:

si $t_{\text{act}} = 1$ **entonces**

valor_alfa = $\alpha(k_{\text{act}}, t=1, \lambda=2)$

si valor_alfa > α^* **entonces**

// Actualizamos $k_{\text{máx}}$ (proposición 4.10):

$k_{\text{máx}} = k_{\text{act}} - 1$ // En general, $k_{\text{máx}} = \min\{k_{\text{máx}}, k_{\text{act}} - 1\}$

si $k_{\text{máx}} = 0$ **entonces**

estado = "FIN – Infactible por Alfa" // Corolario 4.15

sol_act.es_factible = falso

en otro caso

sol_act = OPC_improve_lambda(sol_act) // Algoritmo 5.1

en otro caso

sol_act = OPC_improve_lambda(sol_act) // Algoritmo 5.1

Algoritmo 5.3: Procedimiento OPCenum_update_status

Entrada: Solución actual $\text{sol_act} = (k_{\text{act}}, t_{\text{act}}, \lambda'_{\text{act}})$ y mejor solución encontrada hasta el momento sol_mej .

Salida: Actualiza el valor de sol_mej si corresponde.

si $\text{sol_act.es_factible}$ **entonces**

si estado = "No Infactible" **entonces**

 // Ya se han encontrado soluciones factibles previas con las que comparar.

 // Actualizamos sol_mej si corresponde:

si $f(\text{sol_act}) < f(\text{sol_mej})$ **entonces**

$\text{sol_mej} = \text{sol_act}$

en otro caso

 // Es la primera solución factible que encontramos:

$\text{sol_mej} = \text{sol_act}$

 estado = "No Infactible"

Algoritmo 5.4: Procedimiento OPCenum_set_next_step

Entrada: Solución actual $\text{sol_act} = (k_{\text{act}}, t_{\text{act}}, \lambda'_{\text{act}})$.

Salida: Calcula la nueva solución a explorar (si la hay) y la almacena en sol_act . También actualiza, en su caso, el estado del problema.

// La variable auxiliar global 'continuar' indica en cada momento si existen todavía soluciones pendientes de explorar. Por defecto, toma el valor 'verdadero'.

// Determinamos, si la hay, cuál debe ser la siguiente solución a explorar:

si $k_{\text{act}} \geq k_{\text{máx}}$ **entonces** // Ya se ha terminado con el t actual

si $(k_{\text{máx}} = 0)$ **O** $(t_{\text{act}} \geq T_{\text{máx}})$ **entonces**

 // Hemos terminado de explorar el espacio de soluciones

 continuar = **falso**

si estado = "Sin solución aún" **entonces**

 estado = "FIN – Infactible"

en otro caso

 estado = "FIN – Solución óptima"

en otro caso

 // Cambiamos al siguiente t :

$k_{\text{sig}} = 1$

$t_{\text{sig}} = t_{\text{act}} + 1$

en otro caso

 // Por defecto, la siguiente solución a explorar se construye manteniendo t y aumentando k :

$k_{\text{sig}} = k_{\text{act}} + 1$

$t_{\text{sig}} = t_{\text{act}}$

// Actualizamos el contenido de la variable sol_act :

si continuar = **verdadero** **entonces**

$\text{sol_act} = (k_{\text{sig}}, t_{\text{sig}}, \lambda'_{\text{act}})$ // No es necesario actualizar el valor de λ , porque la nueva solución inmediatamente va a pasar por el procedimiento de mejora de λ

Algoritmo 5.5: Algoritmo OPCenum

Entrada: Funciones integrantes del problema OPC-sb (4.7) f , α y β (véase la sección 3.3) y valor de los parámetros C_p , δ^* , α^* , β^* , $K_{\text{máx}}$ y $T_{\text{máx}}$ (definidos en la subsección 4.1.1).

Salida: Aplica el algoritmo de enumeración implícita descrito en la sección 5.4 y devuelve la solución óptima del problema, si la hay.

```

leer  $f$ ,  $\alpha$ ,  $\beta$  // Funciones
leer  $C_p$ ,  $\delta^*$ ,  $\alpha^*$ ,  $\beta^*$ ,  $K_{\text{máx}}$ ,  $T_{\text{máx}}$  // Parámetros
// Inicialización:
 $k_{\text{máx}} = K_{\text{máx}}$ 
sol_act = ( $k=1, t=1, \lambda=4$ ) // Valor estándar de  $\lambda$  en el PC clásico
sol_mej = vacío
estado = "Sin solución aún"
continuar = verdadero
iter = 0
hacer
    iter = iter + 1
    OPCenum_improve(sol_act) // Algoritmo 5.2
    si estado = "FIN – Infactible por Alfa" entonces
        continuar = falso
    en otro caso
        OPCenum_update_status // Algoritmo 5.3
        OPCenum_set_next_step // Algoritmo 5.4
mientras continuar = verdadero
mostrar estado, iter
si sol_mej  $\neq$  vacío entonces
    mostrar sol_mej

```

5.4.5. Ejemplo numérico

Tras desarrollar formalmente el algoritmo de enumeración implícita OPC.enum, presentamos ahora un ejemplo sencillo con el fin de ilustrar y hacer más claro su funcionamiento.

Concretamente, mostraremos la resolución del problema OPC-sb (4.7) mediante OPC.enum para el caso en que $C_p = 1,2$, $\alpha^* = 0,01$, $\beta^* = 0,1$ y $\delta^* = 1,5462$ (es decir, el mismo supuesto utilizado en la subsección 5.3.1, que además se corresponde con el ejemplo 9 de la batería de problemas usada en la experiencia numérica del capítulo 3; véase también la figura 4.20 del capítulo 4). El valor de los parámetros K_{\max} y T_{\max} será fijado inicialmente a 41.

La ejecución del algoritmo comienza inicializando k_{\max} (máximo valor de k a explorar) al valor de K_{\max} ; es decir, en este caso $k_{\max} = 41$. También se toma como primera solución a explorar $\text{sol}_{\text{act}} = (k = 1, t = 1, \lambda = 4)$, y se inicializan la mejor solución encontrada hasta el momento $\text{sol}_{\text{mej}} = \text{vacío}$ y el estado actual de la resolución 'estado' = «Sin solución aún».

A partir de ahí, el algoritmo comienza a realizar iteraciones siguiendo el esquema detallado en el algoritmo 5.5.

La tabla 5.1 presenta un extracto del resultado de las iteraciones, especificando en cada una de ellas cierta información sobre la solución que está siendo explorada en esa iteración, la mejor solución encontrada hasta ese momento, el estado de la ejecución, y el valor del parámetro k_{\max} en ese momento.

El algoritmo realiza en este ejemplo un total de 1272 iteraciones. En la tabla se observa que las primeras iteraciones no proporcionan ninguna solución factible. En la iteración número 32 se evalúa el par $(k = 32, t = 1)$ y se da el caso de infactibilidad por α ; concretamente, $\alpha(k = 32, t = 1, \lambda = 2) = 0,0101329 > 0,010 = \alpha^*$, y por la proposición 4.10 sabemos que se puede descartar toda solución con $k > 31$, es decir, k_{\max} se actualiza al valor 31.

La primera solución factible aparece en la iteración número 71, con $k = 8$ y $t = 3$, y proporciona un valor de 11,4076 para la función objetivo.

iter	sol _{act}							sol _{mej}				estado	k _{máx}
	k	t	λ'	f	holg. α	holg. β	fact.	k	t	λ'	f		
0	1	1	4	1	-0,0618606	-0,499747	no	—	—	—	—	Sin sol. aún	41
1	1	1	18,0057	1	-0,679250	0	no	—	—	—	—	Sin sol. aún	41
2	2	1	6,66624	1,71989	-0,471761	0	no	—	—	—	—	Sin sol. aún	41
3	3	1	4,93279	2,58767	-0,363646	0	no	—	—	—	—	Sin sol. aún	41
...
31	31	1	2,39175	29,8167	-0,0678052	0	no	—	—	—	—	Sin sol. aún	41
32	32	1	2,39175	29,8167	-0,0678052	0	no	—	—	—	—	Sin sol. aún	31
33	1	2	4,35862	1,09824	0	-0,678839	no	—	—	—	—	Sin sol. aún	31
...
70	7	3	4,24519	10,2804	0	-0,00285772	no	—	—	—	—	Sin sol. aún	31
71	8	3	4,03350	11,4076	0,00209854	0	sí	8	3	4,03350	11,4076	No infact.	31
72	9	3	3,85660	12,4738	0,00329481	0	sí	8	3	4,03350	11,4076	No infact.	31
...
99	5	4	5,31206	9,15589	0	-0,0047606	no	8	3	4,03350	11,4076	No infact.	31
100	6	4	4,84796	10,3265	0,00355727	0	sí	6	4	4,84796	10,3265	No infact.	31
101	7	4	4,49988	11,3807	0,00493306	0	sí	6	4	4,84796	10,3265	No infact.	31
...
129	4	5	6,34545	8,77140	0	-0,0185529	no	6	4	4,84796	10,3265	No infact.	31
130	5	5	5,63214	10,2470	0,00406504	0	sí	5	5	5,63214	10,2470	No infact.	31
131	6	5	5,06798	11,2427	0,00556996	0	sí	5	5	5,63214	10,2470	No infact.	31

Tabla 5.1: Detalle de las iteraciones correspondientes al ejemplo de aplicación del algoritmo OPCenum propuesto en la subsección 5.4.5. Para cada iteración 'iter' se detalla la solución que se ha explorado en ella sol_{act} ya mejorada (con $\lambda'(k, t)$), así como las holguras de ésta respecto de las dos restricciones y si es o no factible; también se indica la mejor solución hallada hasta ese momento, el estado de la ejecución del algoritmo, y el valor de k_{máx} tras esa iteración. En negrita se destacan las iteraciones en que se producen cambios en la mejor solución o en k_{máx}. *Continúa en la página siguiente.*

iter	sol _{act}							sol _{mej}				estado	k _{máx}
	k	t	λ'	f	holg. α	holg. β	fact.	k	t	λ'	f		
...
159	3	6	7,58359	7,27638	0	-0,087576	no	5	5	5,63214	10,2470	No infact.	31
160	4	6	6,87310	10,2431	0,00202086	0	sí	4	6	6,87310	10,2431	No infact.	31
161	5	6	5,86215	11,1163	0,00550354	0	sí	4	6	6,87310	10,2431	No infact.	31
...
314	3	11	10,8744	14,4270	0,00131297	0	sí	4	6	6,87310	10,2431	No infact.	31
315	4	11	7,93086	13,9765	0,00542166	0	sí	4	6	6,87310	10,2431	No infact.	31
316	5	11	6,47121	13,8164	0,00559867	0	sí	4	6	6,87310	10,2431	No infact.	31
317	6	11	5,58507	13,8213	0,00560162	0	sí	4	6	6,87310	10,2431	No infact.	31
318	7	11	4,99523	14,0262	0,00553661	0	sí	4	6	6,87310	10,2431	No infact.	31
...
1269	28	41	2,72929	31,3707	0,0000173046	0	sí	4	6	6,87310	10,2431	No infact.	31
1270	29	41	2,61632	31,4251	0	-0,0494847	no	4	6	6,87310	10,2431	No infact.	31
1271	30	41	2,47351	31,4251	0	-0,136140	no	4	6	6,87310	10,2431	No infact.	31
1272	31	41	2,26416	31,4251	0	-0,284584	no	4	6	6,87310	10,2431	FIN - S.Ópt.	31

Tabla 5.1: Detalle de las iteraciones correspondientes al ejemplo de aplicación del algoritmo OPCenum propuesto en la subsección 5.4.5. Para cada iteración 'iter' se detalla la solución que se ha explorado en ella sol_{act} ya mejorada (con $\lambda'(k, t)$), así como las holguras de ésta respecto de las dos restricciones y si es o no factible; también se indica la mejor solución hallada hasta ese momento, el estado de la ejecución del algoritmo, y el valor de k_{máx} tras esa iteración. En negrita se destacan las iteraciones en que se producen cambios en la mejor solución o en k_{máx}.

Nótese que *no* podemos asegurar que la mejor solución para $k = 9$ y $t = 3$ (que es la siguiente iteración) vaya a ser necesariamente *peor* que la anterior (por tener la misma t y una k mayor), ya que el valor de λ' varía de una a otra solución. De hecho, puede observarse cómo en la iteración 314 el mínimo valor que se obtiene para f , fijadas k a 3 y t a 11, es 14,4270, mientras que en la siguiente iteración, con $k = 4$ y $t = 11$, se obtiene una solución con f igual a 13,9765; y sigue disminuyendo con $k = 5$ y $t = 11$ (f toma un valor de 13,8164), hasta que ya en la siguiente iteración comienza a aumentar. Ya se había advertido en la subsección 5.4.2 la posible no monotonía de $f(k, t, \lambda'(k, t))$ para valores crecientes de k , fijado el parámetro t .

La siguiente mejora del valor de la función objetivo, dentro de la región factible del problema estudiado, se produce en la iteración número 100, con $k = 6$ y $t = 4$ y un valor para f de 10,3265 (un 9,5% mejor que el anterior). Posteriormente se encuentra otra mejora en la iteración 130, para el par $(k = 5, t = 5)$, que devuelve un valor de f de 10,2470 (se corresponde con el ejemplo 1 de la experiencia numérica 5.3.1). La solución que al final resulta ser la óptima se alcanza en la iteración 160; es $(k = 4, t = 6, \lambda = 6,87310)$ y proporciona un mínimo para f de 10,2431 (un 10,2% mejor que la primera solución factible encontrada).

Tras evaluar el caso con $k = 31$ y $t = 41$ la exploración de la región factible se da por terminada, y el algoritmo informa, en este caso, que ha resuelto de manera óptima el problema OPC-sb (4.7), devolviendo la solución mencionada; es decir:

$$(k^* = 4, t^* = 6, \lambda^* = 6,87310), \quad f^* = 10,2470.$$

Por tanto, teniendo en cuenta la relación entre λ y W_G (2.24), podemos concluir que el *plan de precontrol óptimo* consistirá en definir la anchura de la zona verde como $(USL - LSL)/(6,87310/2)$ (es decir, un 29,10% de la anchura del rango de tolerancias), cualificando el proceso cuando se observen 4 piezas verdes consecutivas y no cualificándolo cuando se contabilicen 6 amarillas consecutivas o bien se muestree una roja. Con

ello se asegura un tamaño medio de muestreo de 10,2470 piezas en el caso en que el proceso esté centrado, además de una tasa de falsa alarma de $0,01 - 0,00202086 = 0,00797914$ y una potencia de $1 - 0,1 = 0,9$ para detectar desviaciones del proceso del orden de 1,562 desviaciones típicas o mayores.

Como ya se ha mencionado, este ejemplo que terminamos de estudiar se corresponde con el problema 9 de la batería utilizada para las pruebas numéricas llevadas a cabo con GAMS y Excel Solver en el capítulo 3 (véase la tabla A.1 del anexo A). En aquella experiencia numérica, de las 135 ejecuciones con diferentes algoritmos sobre este mismo problema, sólo 65 (el 48,1%) alcanzaron la verdadera solución óptima; otras 14 ejecuciones (10,4%) devolvieron la solución obtenida en la iteración 130, con $k = 5$ y $t = 5$; 3 devolvieron la solución obtenida en la iteración número 100, con $k = 5$ y $t = 4$, o la obtenida en la iteración número 71, con $k = 8$ y $t = 3$; otras 32 ejecuciones devolvieron otras soluciones con un valor de f peor que las anteriores; y el resto (21 ejecuciones; un 15,6%) no devolvieron ninguna solución válida.

5.5. Implementación de los algoritmos⁴

El algoritmo OPCenum y todos los procedimientos y funciones que engloba (OPC_improve_lambda, OPCenum_improve, OPCenum_update_status y OPCenum_set_next_step), descritos en las secciones 5.3 y 5.4, se implementaron mediante el lenguaje de programación C++. En concreto, se utilizó el entorno de programación Microsoft Visual Studio 2008 (versión 9.0.30729.1 SP).

Se valoró la implementación del algoritmo en otros lenguajes o sistemas, tales como Python, Mathematica, Matlab, etc. Se eligió C++ por su madurez, estabilidad y versatilidad; por la abundancia de bibliotecas

⁴En toda esta sección, utilizaremos el término *variable* para referirnos tanto a las variables de decisión del problema OPC, k , t y λ , como a las variables en el sentido de programación informática, es decir, a las posiciones de memoria identificadas mediante un nombre que contienen cierta cantidad de información, en principio susceptible de ser modificada.

de funciones y otros recursos de calidad contrastada elaborados por terceros para dicho lenguaje —algunos de los cuales mencionaremos posteriormente—; y por el hecho de ser un lenguaje compilado, lo cual permite obtener ejecuciones eficientes —aunque en nuestro caso, finalmente, esto no sea del todo relevante, dada la relativa sencillez del algoritmo a implementar—. No obstante, consideramos que nuestro algoritmo podría haberse programado con éxito en cualquier otro de los entornos alternativos mencionados. De hecho, una de las líneas futuras de trabajo que detallaremos en el capítulo 7 de esta memoria conlleva su implementación en Mathematica.

5.5.1. Detalles de la implementación

Si bien C++ es un lenguaje orientado a objetos, a la hora de programar nuestro algoritmo no se hizo uso de toda la funcionalidad disponible en tal aspecto. Se optó, en cambio, por un enfoque de programación clásico, basado en procedimientos.

De manera coherente con el paradigma de programación por procedimientos, y con el fin de facilitar la reutilización del código en futuros trabajos, se definieron tipos de datos compuestos (conocidos como *structs* en inglés) para representar y manejar de manera encapsulada entidades tales como

- el conjunto de parámetros o condiciones iniciales del problema a resolver;
- el conjunto de cotas inferiores y superiores establecidas sobre las variables de decisión;
- las restricciones;
- la función objetivo;
- una solución del problema y toda la información relevante asociada:
 - valor de las variables k , t y λ ,

- valor de la función objetivo para dicha solución,
 - si es o no factible,
 - holgura de dicha solución respecto de cada una de las restricciones,
 - si la solución tiene holgura cero o no respecto de alguna de las restricciones,
 - los valores de λ_α y λ_β para ese par (k, t) y
 - el valor de la función objetivo para (k, t, λ_α) y (k, t, λ_β) .
- un problema de OPC, que contiene la siguiente información:
- información relativa al problema a resolver y que, para un problema dado, una vez inicializada no se modifica:
 - condiciones iniciales,
 - información sobre las variables tales como cotas inferiores y superiores,
 - restricciones y
 - función objetivo;
 - información que maneja el algoritmo durante su ejecución, tales como:
 - la mejor solución encontrada hasta el momento sol_{mej} ,
 - la solución que está siendo evaluada en el momento actual sol_{act} ,
 - el estado actual de la resolución (si se ha encontrado ya o no alguna solución factible, si se ha resuelto de manera óptima o si es un problema imposible; variable 'estado' en los algoritmos en pseudocódigo mostrados en la sección 5.4) y
 - un indicador de si la ejecución ha terminado o si hay que continuar (variable 'continuar' en los algoritmos en pseudocódigo);

- estructuras de memoria del algoritmo: en nuestro caso, el único dato que puede considerarse como tal es el parámetro $k_{\text{máx}}$; y
- contadores auxiliares del número de llamadas a las funciones α , β y f durante una ejecución concreta del algoritmo sobre este problema.

Nótese que las variables tratadas como globales en los algoritmos descritos en pseudocódigo en la sección 5.4, tales como sol_{mej} , sol_{act} , 'estado', 'continuar' y $k_{\text{máx}}$ fueron incluidas dentro de la estructura que contiene toda la información del problema OPC y del algoritmo, la cual acabamos de describir. De ese modo, lo que se hizo para que los procedimientos y funciones pudieran *ver* esas variables fue incluir como argumento de dichas subrutinas la variable que contenía toda la información del problema.

Además de las subrutinas que forman parte del algoritmo en sí mismas, se definieron otras para realizar operaciones básicas tales como el cálculo de las probabilidades de pieza verde y amarilla, el cálculo de las probabilidades α y β y de la función objetivo f , el cálculo de holguras, etc., u otras auxiliares, tales como la inicialización de algunas variables de tipo compuesto, la lectura y la presentación de datos, etc.

En particular, se definieron ciertos tipos de datos compuestos y subrutinas auxiliares para poder conectar nuestro algoritmo con las subrutinas para el cálculo de raíces pertenecientes a la Biblioteca Científica GNU, y que describimos con más detalle en la subsección 5.5.2.

También se desarrolló una subrutina para obtener un intervalo inicial válido previo a cada ejecución del método de búsqueda de raíces, basada en una estrategia sencilla de búsqueda incremental, como se ha dicho en la sección 5.3.

Tal y como se deduce de la estructura de datos presentada anteriormente, se implementó un conteo acumulado del número de llamadas a las funciones α , β y f para cada ejecución de OPCenum . Los contadores se van actualizando ante *cualquier* llamada a dichas funciones, las cuales se producen principalmente al aplicar el método de búsqueda de raíces y

en la subrutina auxiliar de búsqueda incremental (es decir, dentro del algoritmo auxiliar de mejora local 5.1).

Se definió una subrutina para medir el tiempo de uso de la CPU de una ejecución del algoritmo⁵.

Todas las variables de naturaleza continua y todos los cálculos asociados se almacenaron con precisión doble⁶.

En cuanto al método de búsqueda de raíces, se programó para que en cada llamada iterarse hasta alcanzar un error absoluto residual de $1 \cdot 10^{-14}$, es decir, hasta que $|\alpha(\lambda | k_0, t_0) - \alpha^*| < 1 \cdot 10^{-14}$ o, en su caso, $|\beta(\lambda | k_0, t_0, \delta^*) - \beta^*| < 1 \cdot 10^{-14}$ ⁷.

Cuando se consideró posible y conveniente se utilizaron punteros (variables que *apuntan* a la dirección de memoria de otra variable) y se pasaron por referencia los argumentos a las subrutinas (de modo que no recibiesen una *réplica* temporal de las variables que forman parte de los argumentos sino la dirección de memoria de las mismas), con el fin de conseguir una implementación más eficiente.

Se programó una rutina principal que, mediante una serie de bucles anidados, hacía posible resolver en una sola ejecución toda una batería de ejemplos predeterminada (véase más adelante, en la sección 5.6). También se implementó la opción de permitir al usuario introducir un ejemplo concreto.

El algoritmo se programó como un proyecto de C++ de tipo consola (es decir, sin interfaz gráfica de ventanas, botones, etc.). Tanto la lectura de datos (en el caso de la introducción de un ejemplo concreto) como la

⁵Se hizo uso de la función `GetProcessTimes()` disponible a través del archivo de cabecera `windows.h`.

Existe una alternativa más conocida, e independiente de la plataforma, consistente en usar la función `clock()` de la librería C `Time (time.h)`, que, de acuerdo a los estándares de C y C++, debería devolver tiempo de CPU, pero se sabe que en Visual Studio 2008 dicha función realmente trabaja con el tiempo real de ejecución o *wall-clock time*.

En cualquier caso, la importancia de este detalle es relativa, dados los tiempos de ejecución que se espera tener que manejar.

⁶Concretamente, se utilizó el tipo de datos `double` de C++. El estándar de C y C++ no predetermina la longitud de un dato definido como `double`. En Microsoft C++ se le asigna una longitud de 8 bytes, que equivale en la práctica a 15 dígitos significativos.

⁷Para ello se utilizó la función de la GSL `gsl_root_test_residual`.

presentación de resultados se realizó a través de archivos de texto plano. Como trabajos futuros (véase el capítulo 7) se considera la elaboración de una versión del programa con una interfaz gráfica fácil de usar.

El código fuente del programa está disponible bajo petición al autor.

5.5.2. Uso de bibliotecas externas

Como ya se ha dicho en la sección 5.3, se eligió el método de Brent como algoritmo auxiliar para la determinación de los valores λ_α y λ_β , dentro del algoritmo de mejora local.

Este es un paso fundamental dentro del algoritmo OPCenum que aquí hemos presentado. Por ello, y dado que el objetivo de nuestro trabajo no es el desarrollo ni mejora de métodos numéricos para la búsqueda de raíces, decidimos usar una implementación del método de Brent programada por terceros, depurada y probada por una amplia comunidad de usuarios y disponible de manera gratuita para los lenguajes C y C++: la incluida en la Biblioteca Científica GNU (*GNU Scientific Library, GSL*; Galassi et al., 2009, Free Software Foundation, Inc., 2009). Esta biblioteca, que goza de buena reputación, existe desde 1996, y ofrece una completa colección de métodos numéricos agrupados por temáticas: estadística, interpolación, optimización, generación de números aleatorios, búsqueda de raíces, etc. La versión de la librería que se utilizó es la 1.13, de septiembre de 2009.

La GSL contiene un conjunto de subrutinas relacionadas con los diferentes pasos comunes a todos los procedimientos de búsqueda de raíces de una función: inicializar la búsqueda, realizar una iteración del procedimiento, devolver el punto candidato tras la última iteración, devolver el intervalo de búsqueda tras la última iteración (en los métodos de acotación), determinar si se cumple el criterio de parada del método, etc. En el proceso de inicialización se indica convenientemente el algoritmo a utilizar (en nuestro caso, el método de Brent) así como la función de la cual se desea obtener una raíz⁸. En nuestro código, fue necesario realizar una —sencilla— adaptación de las funciones α y β para adecuarlas al

⁸Todas estas subrutinas y funcionalidades están accesibles en un programa de C++ mediante la inclusión del archivo de cabecera `gsl_roots.h`.

formato genérico de *función* que requieren las diferentes subrutinas de la GSL.

En la implementación de nuestro algoritmo OPCenum hicimos uso también de otras subrutinas de la GSL. Concretamente, dentro de las funciones diseñadas para el cálculo de las probabilidades de muestrear una pieza verde y amarilla, utilizamos las funciones de la GSL que permiten obtener probabilidades asociadas a la distribución normal tipificada⁹. De nuevo, la idea fue confiar en una biblioteca externa, de calidad reconocida, para realizar dichos cálculos, que son esenciales para que el algoritmo proporcione resultados fiables.

5.6. Experiencia numérica

El algoritmo OPCenum (algoritmo 5.5), implementado como se detalla en la sección 5.5, fue ejecutado en la misma máquina y sobre la misma batería de ejemplos que los empleados en la experiencia numérica con GAMS y Excel Solver (subsección 3.5.2).

Recordemos que se trataba de una colección de 36 ejemplos, resultantes de combinar los siguientes valores para los parámetros del problema: $C_p \in \{1,2; 1,3333; 1,4; 1,5\} \times \alpha^* \in \{0,27\%; 0,5\%; 1\%\} \times \beta^* \in \{0,5\%; 5\%; 10\%\}$, con δ^* tal que $p^* = 2\%$ (véase la tabla A.1 del anexo A).

En esta ocasión, tal y como se ha justificado al comienzo del capítulo, las pruebas se realizaron sobre el problema OPC-sb (4.7), con $K_{\text{máx}} = T_{\text{máx}} = 41$ (imitando la elección de estas cotas realizada en la subsección 3.5.2); no se limitó, por tanto, el valor de la variable λ , a diferencia de las pruebas con GAMS y Solver.

⁹Las funciones de la GSL asociadas a distribuciones de probabilidad conocidas están accesibles mediante el archivo de cabecera `gsl_cdf.h`.

5.6.1. Resultados numéricos

El tiempo que duró la ejecución de las 36 pruebas fue de 2,62 segundos (tiempo de CPU). El estado final tras resolver cada ejemplo fue «*FIN - Infactible*» o «*FIN - Solución óptima*» (véanse los algoritmos 5.2, 5.3 y 5.4). En concreto, los ejemplos 1 a 5 de la batería fueron identificados como imposibles por OPCenum, y el resto como factibles.

Las *soluciones óptimas* que devolvió OPCenum en aquellos ejemplos factibles fueron, en *todos* los casos, idénticas a las respectivas mejores soluciones encontradas en la experiencia numérica del capítulo 3. En la subsección 5.6.2 ampliamos la comparación de los resultados obtenidos en ambas experiencias.

En cualquier caso, cabe recordar y destacar que, por el modo en como se ha diseñado el algoritmo, éste se comporta de modo *exacto* (y no heurístico) sobre estos 36 y sobre cualquier otro ejemplo del problema OPC-sb (4.7). Por ello puede afirmarse con propiedad que las soluciones obtenidas son *óptimas*.

En la tabla 5.2 se detallan el número de llamadas a las funciones del problema y el número de iteraciones que necesitó OPCenum para resolver cada uno de los 36 ejemplos de la batería, así como el valor final del parámetro $k_{\text{máx}}$ en cada caso.

Por el diseño del algoritmo en sí, el número de iteraciones está directamente relacionado con el valor que toma el parámetro $k_{\text{máx}}$ cuando se terminan de explorar las soluciones con $t = 1$. Así, en los ejemplos 1 a 3 (todos con $C_p = 1,2$ y $\alpha = 0,0027$) el valor máximo de k a explorar terminó siendo $k_{\text{máx}} = 8$ (es decir, no fue necesario explorar ninguna solución con $k > 8$ ya que, por la proposición 4.10, se puede demostrar que todas ellas incumplirán la restricción sobre α), lo cual se tradujo finalmente en 329 iteraciones (es decir, el algoritmo exploró 329 pares (k, t)): 9 con $t = 1$ (es tras explorar el par $(k = 9, t = 1)$ cuando se actualiza el valor de $k_{\text{máx}}$ a 8) y 8 para $t = 2, \dots, 41$.

Para los ejemplos 4 a 6 (con $C_p = 1,2$ y $\alpha = 0,005$) el valor de $k_{\text{máx}}$ terminó siendo 15, y el número de iteraciones que realizó OPCenum en todos ellos fue $16 + 15 \cdot 40 = 616$.

N.º ej.*	Parámetros del problema				Comportamiento de OPCenum					
	C_p	α^*	β^*	$\delta^{*\dagger}$	N.II. α^\ddagger	N.II. β^{\S}	N.II. f^{\parallel}	Estado	Núm. iter.**	$k_{\text{máx}}^{\text{++}}$
1	1,2	0,0027	0,005	1,5462	4100	5465	657	Imp.	329	8
2	1,2	0,0027	0,05	1,5462	4098	4882	657	Imp.	329	8
3	1,2	0,0027	0,1	1,5462	4100	4595	657	Imp.	329	8
4	1,2	0,005	0,005	1,5462	7401	9483	1231	Imp.	616	15
5	1,2	0,005	0,05	1,5462	7402	8343	1231	Imp.	616	15
6	1,2	0,005	0,1	1,5462	7323	8313	1231	S.Ópt.	616	15
7	1,2	0,01	0,005	1,5462	14944	17747	2543	S.Ópt.	1272	31
8	1,2	0,01	0,05	1,5462	15813	16132	2543	S.Ópt.	1272	31
9	1,2	0,01	0,1	1,5462	16353	14964	2543	S.Ópt.	1272	31
10	1,3333	0,0027	0,005	1,9463	22570	21402	3363	S.Ópt.	1681	41
11	1,3333	0,0027	0,05	1,9463	22169	19527	3363	S.Ópt.	1681	41
12	1,3333	0,0027	0,1	1,9463	22828	18659	3363	S.Ópt.	1681	41
13	1,3333	0,005	0,005	1,9463	21982	21143	3363	S.Ópt.	1681	41
14	1,3333	0,005	0,05	1,9463	23601	19528	3363	S.Ópt.	1681	41
15	1,3333	0,005	0,1	1,9463	24212	18666	3363	S.Ópt.	1681	41
16	1,3333	0,01	0,005	1,9463	24237	21118	3363	S.Ópt.	1681	41
17	1,3333	0,01	0,05	1,9463	26206	19528	3363	S.Ópt.	1681	41
18	1,3333	0,01	0,1	1,9463	25899	18680	3363	S.Ópt.	1681	41
19	1,4	0,0027	0,005	2,1463	24113	21165	3363	S.Ópt.	1681	41
20	1,4	0,0027	0,05	2,1463	25909	19460	3363	S.Ópt.	1681	41
21	1,4	0,0027	0,1	2,1463	25447	18623	3363	S.Ópt.	1681	41
22	1,4	0,005	0,005	2,1463	26260	21155	3363	S.Ópt.	1681	41
23	1,4	0,005	0,05	2,1463	26277	19475	3363	S.Ópt.	1681	41
24	1,4	0,005	0,1	2,1463	27184	18630	3363	S.Ópt.	1681	41
25	1,4	0,01	0,005	2,1463	27855	21140	3363	S.Ópt.	1681	41
26	1,4	0,01	0,05	2,1463	28465	19495	3363	S.Ópt.	1681	41
27	1,4	0,01	0,1	2,1463	28426	18636	3363	S.Ópt.	1681	41

Tabla 5.2: Resultados obtenidos al aplicar el algoritmo OPCenum sobre la misma batería de ejemplos utilizada en la experiencia numérica de la sección 3.5.2. Se omite la solución óptima hallada por ser idéntica, en todos los ejemplos no infactibles, a la respectiva mejor solución obtenida en aquella experiencia numérica (véase la tabla A.1 del anexo A). Para cada problema, se indica el número de llamadas realizadas a las funciones α , β y f , así como el número de iteraciones realizadas y el valor final del parámetro $k_{\text{máx}}$. *Continúa en la página siguiente.*

N.º ej.*	Parámetros del problema				Comportamiento de OPCenum					
	C_p	α^*	β^*	$\delta^{*\dagger}$	N.ll. α^\ddagger	N.ll. β^\S	N.ll. f^\P	Estado	Núm. iter.**	$k_{\text{máx}}^{\text{++}}$
28	1,5	0,0027	0,005	2,4463	28600	21145	3363	S.Ópt.	1681	41
29	1,5	0,0027	0,05	2,4463	27722	19426	3363	S.Ópt.	1681	41
30	1,5	0,0027	0,1	2,4463	27710	18584	3363	S.Ópt.	1681	41
31	1,5	0,005	0,005	2,4463	29928	21146	3363	S.Ópt.	1681	41
32	1,5	0,005	0,05	2,4463	30346	19451	3363	S.Ópt.	1681	41
33	1,5	0,005	0,1	2,4463	30650	18637	3363	S.Ópt.	1681	41
34	1,5	0,01	0,005	2,4463	32925	21149	3363	S.Ópt.	1681	41
35	1,5	0,01	0,05	2,4463	33768	19519	3363	S.Ópt.	1681	41
36	1,5	0,01	0,1	2,4463	33861	18687	3363	S.Ópt.	1681	41

* Número identificador de ejemplo.

† El valor del parámetro δ^* , desviación inaceptable del proceso, viene fijado por el de C_p , de manera que la proporción de piezas fuera de especificaciones bajo dicha desviación sea el 2%; véase la expresión (2.34).

‡ Número de llamadas a la función α .

§ Número de llamadas a la función β .

¶ Número de llamadas a la función f .

|| Estado final del algoritmo: Imp. = Problema imposible. S.Ópt. = Solución óptima hallada.

** Número total de iteraciones realizadas.

++ Valor final del parámetro $k_{\text{máx}}$ tras la ejecución del algoritmo.

Tabla 5.2: Resultados obtenidos al aplicar el algoritmo OPCenum sobre la misma batería de ejemplos utilizada en la experiencia numérica de la sección 3.5.2. Se omite la solución óptima hallada por ser idéntica, en todos los ejemplos no infactibles, a la respectiva mejor solución obtenida en aquella experiencia numérica (véase la tabla A.1 del anexo A). Para cada problema, se indica el número de llamadas realizadas a las funciones α , β y f , así como el número de iteraciones realizadas y el valor final del parámetro $k_{\text{máx}}$.

En los ejemplos 7 a 9 ($C_p = 1,2$ y $\alpha = 0,01$) el parámetro $k_{\text{máx}}$ tomó el valor 31 al final de la ejecución del algoritmo, con un total de $32 + 31 \cdot 40 = 1272$ iteraciones en cada caso.

Para el resto de ejemplos (del 10 al 36), $k_{\text{máx}}$ terminó valiendo lo mismo que al principio, es decir 41, lo cual significa que en ninguna de estas ejecuciones se aplicó la reducción de la zona de búsqueda debida a infactibilidad por α , y el algoritmo realizó por tanto $41 \cdot 41 = 1681$ iteraciones (el máximo posible, para las condiciones iniciales elegidas

de $K_{\text{máx}} = T_{\text{máx}} = 41$).

El número de evaluaciones de las funciones del problema que fue necesario realizar en cada ejecución de OPCenum está directamente relacionado con el número de iteraciones. Así, en el total de las 36 ejecuciones, se realizaron una media de 15,11 y 12,33 evaluaciones de α y β , respectivamente, por cada iteración, y 2,00 evaluaciones de f por iteración, siendo la dispersión en los tres casos relativamente pequeña.

De hecho, en el caso de f , el número de evaluaciones está directamente relacionado con el número de iteraciones. En una iteración normal, el algoritmo OPCenum realiza dos llamadas a f : una para calcular $f(k = k_0, t = t_0, \lambda = \lambda_\alpha)$ y otra para $f(k = k_0, t = t_0, \lambda = \lambda_\beta)$. Además, el algoritmo realiza una llamada a f para calcular el valor de la función objetivo de la solución inicial, que en el fondo sería prescindible. Por otro lado, en las iteraciones en que se detecta infactibilidad por α , no se llega a evaluar f . De ahí que el número de evaluaciones de f por iteración *no* sea *exactamente* constante e igual a 2 en cada ejecución.

En la tablas 5.3 y 5.4 se detallan los valores medios y las desviaciones típicas de las tasas de evaluaciones de α y β por iteración, respectivamente, en función del valor de los parámetros C_p , α^* y β^* . Se muestran las diferencias más significativas encontradas al realizar diversos contrastes de hipótesis, agrupando valores en su caso. Las figuras 5.3 y 5.4 complementan toda esta información. En ellas se muestran diagramas de caja y bigotes para cada uno de los resultados objeto de interés.

A partir del análisis de la información generada por la experiencia numérica, resumida en las tablas y figuras mencionadas, puede concluirse que el algoritmo OPCenum necesitó evaluar más veces la función α por cada iteración cuanto mayor (es decir, mejor) era la capacidad potencial del proceso C_p (concretamente, un 48,5% más de llamadas a α por iteración con $C_p = 1,5$ que con $C_p = 1,2$, en promedio).

En cambio, el número de evaluaciones de β por iteración fue significativamente mayor en los ejemplos con $C_p = 1,2$ que en el resto (un 19,3% más en promedio), así como en los ejemplos con un valor más restrictivo para β^* (un 14,6% más de llamadas por iteración con $\beta^* = 0,005$ frente a

		n	Media	Desv. típ.	p-valor
Llamadas a α /iteración		36	15,1137	2,41354	
C_p	1,2	9	12,2594	0,357868	< 0,0001
	1,3333	9	14,1255	0,916551	
	1,4	9	15,8593	0,864852	
	1,5	9	18,2107	1,44540	
α^*	0,0027	12	14,3716	1,69205	< 0,0001
	0,005	12	14,9127	2,37743	
	0,01	12	16,0569	2,89898	
β^*	0,005	12	14,8406	2,49308	0,0018
	0,05; 0,1	24	15,2503	2,41524	

Tabla 5.3: Tasa de llamadas a la función α por cada iteración del algoritmo OPCenum, en función de los parámetros del problema, según la experiencia numérica de la sección 5.6. Se indica si la diferencia es significativa de acuerdo a un ANOVA de factores cruzados. Véase también la figura 5.3.

		n	Media	Desv. típ.	p-valor
Llamadas a β /iteración		36	12,3273	1,33031	
C_p	1,2	9	14,0276	1,44309	< 0,0001
	el resto	27	11,7605	0,638113	
α^*	0,0027	12	12,6084	1,73312	< 0,0001
	0,005	12	12,3519	1,30201	
	0,01	12	12,0216	0,862091	
β^*	0,005	12	13,2767	1,35684	< 0,0001
	0,05	12	12,1178	1,05435	
	0,1	12	11,5874	1,02461	

Tabla 5.4: Tasa de llamadas a la función β por cada iteración del algoritmo OPCenum, en función de los parámetros del problema, según la experiencia numérica de la sección 5.6. Se indica si la diferencia es significativa de acuerdo a un ANOVA de factores cruzados. Véase también la figura 5.4.

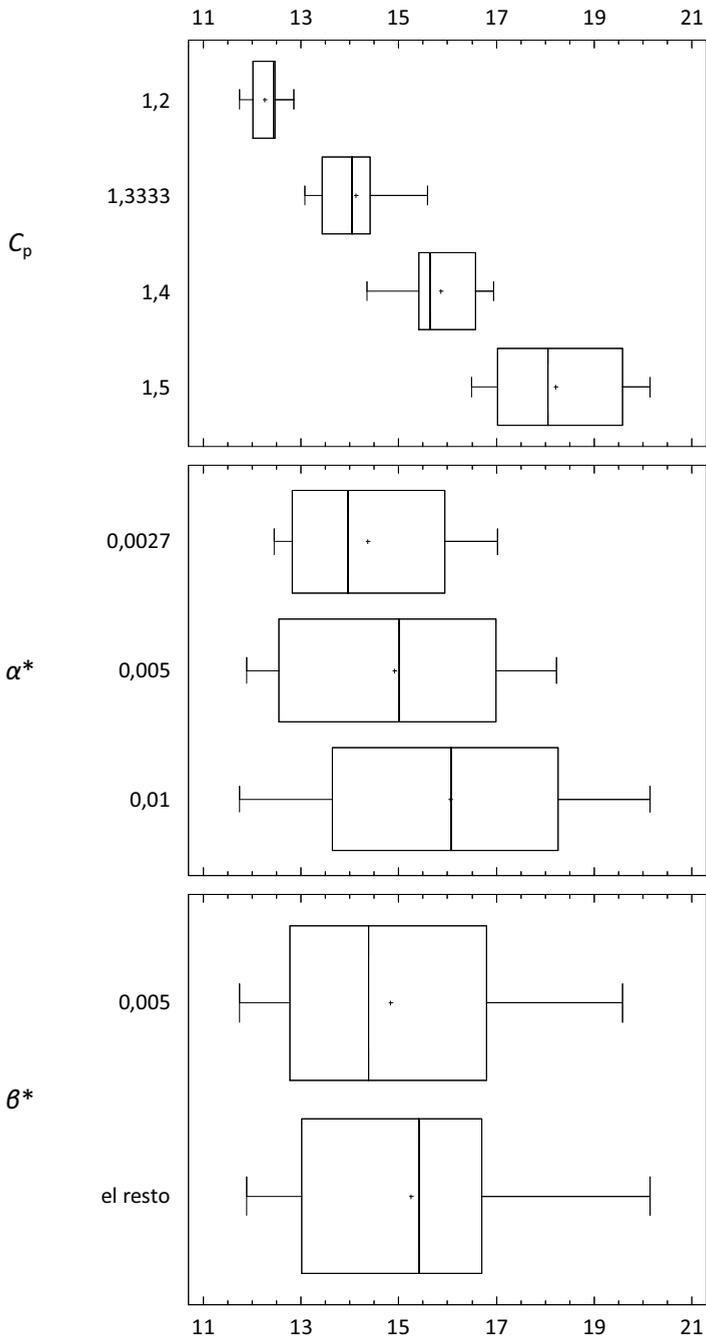


Figura 5.3: Tasa de llamadas a la función α por cada iteración del algoritmo OPCenum, en función de los parámetros del problema, según la experiencia numérica de la sección 5.6.

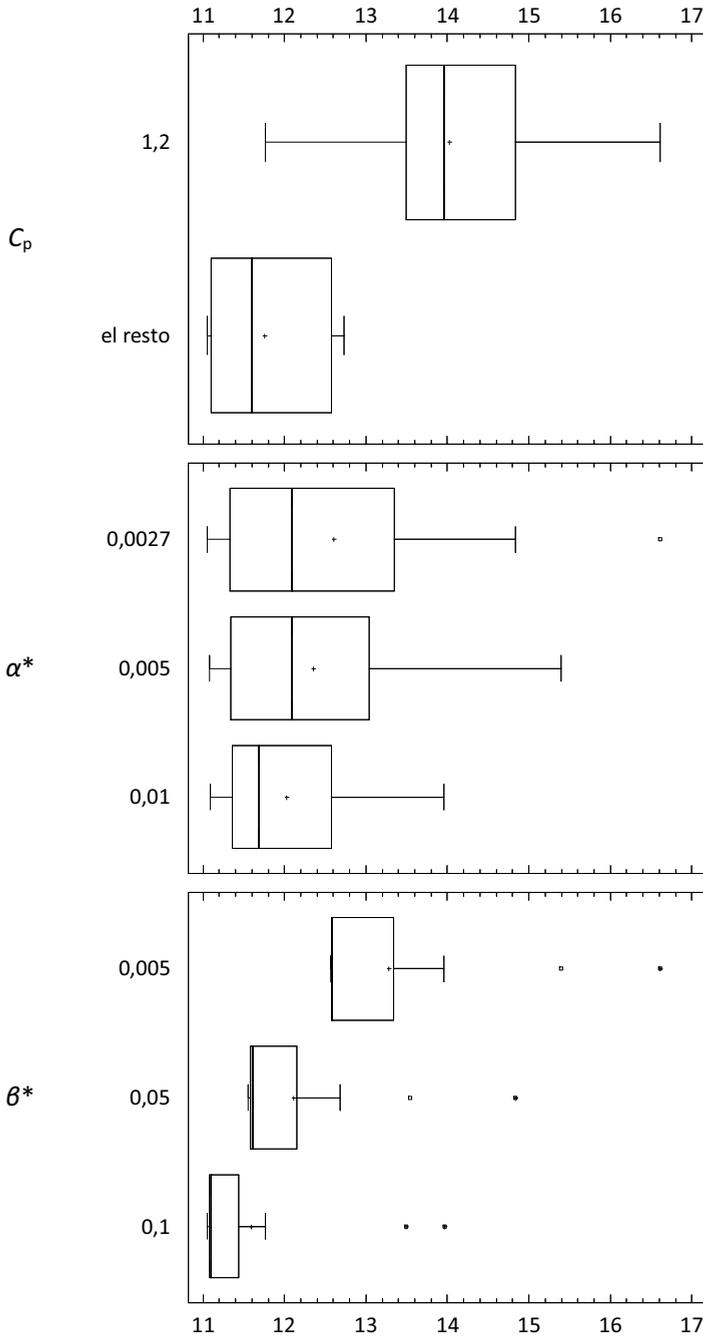


Figura 5.4: Tasa de llamadas a la función β por cada iteración del algoritmo OPCenum, en función de los parámetros del problema, según la experiencia numérica de la sección 5.6.

$\beta^* = 0,1$).

Como se ha explicado a lo largo del presente capítulo y se ha puesto de manifiesto, en concreto, al describir los algoritmos propuestos, la carga computacional recae en la función `OPC_improve_lambda` (algoritmo 5.1), y en concreto en el procedimiento de búsqueda de raíces. También se realizan llamadas a α y β en la función auxiliar para el cálculo de holguras mencionada anteriormente, las cuales en realidad serían prescindibles.

Obviamente, los parámetros del problema influyen en la cantidad de pasos que es necesario realizar en cada llamada al método de Brent (o cualquier otro que se hubiera utilizado), la cual está directamente relacionada con el número de evaluaciones de α y β . Haría falta un análisis más detallado para explicar el porqué de las diferencias concretas observadas. Sin embargo, dada la mínima influencia práctica de dichas diferencias (todas las ejecuciones se realizaron en un tiempo más que aceptable), dicho análisis no parece imprescindible.

5.6.2. Comparación con los resultados de aplicar de *software* de propósito general¹⁰

Como ya se ha destacado, el algoritmo `OPCenum` consigue resolver de manera *exacta* los 36 ejemplos del problema OPC-sb (4.7), y ello, además, permite confirmar los resultados obtenidos mediante la aplicación de *software* de propósito general (véase la subsección 3.5.2).

En concreto, se corrobora que los ejemplos 1 a 5 de la batería son imposibles, y que las mejores soluciones encontradas por GAMS y Solver son efectivamente las óptimas en cada uno de los ejemplos 6 a 36. Es

¹⁰Las experiencias numéricas con *software* de optimización de propósito general llevadas a cabo en el capítulo 3 se efectuaron sobre el problema OPC-b (4.6), mientras que las realizadas con `OPCenum` tuvieron como objeto el problema OPC-sb (4.7). Ambos problemas, como ya se ha visto, sólo difieren en la existencia o ausencia de una cota superior para la variable de decisión λ , fijada en el caso de las experiencias del capítulo 3 a un valor de 42. El hecho de que las mejores soluciones encontradas en aquel capítulo—confirmadas como soluciones óptimas ahora—presenten valores de λ bastante alejados de dicha cota superior hace poco relevante esa diferencia entre ambos problemas y, por tanto, valida la comparación de los resultados obtenidos mediante uno y otro enfoque.

importante recordar que los algoritmos utilizados en aquella experiencia no lograban garantizar la optimalidad de la solución final proporcionada, ni la no factibilidad de aquellos ejemplos para los que no se encontró ninguna solución. Por otro lado, el tiempo observado de ejecución de OPCenum es similar al de los algoritmos generales, y totalmente aceptable en cualquier caso.

Como se ha visto, el comportamiento de OPCenum sobre todos los ejemplos de la batería es más o menos homogéneo, a diferencia de los resultados obtenidos con GAMS y Solver, en los que el porcentaje de ejecuciones que alcanzaban la mejor solución en cada ejemplo variaba desde el 38 hasta el 87 %.

En concreto, en aquella experiencia numérica, los ejemplos factibles con $C_p = 1,2$ (problemas 6 a 9) presentaban un comportamiento diferente al resto, motivo por el cual fueron estudiados a parte. Ahora se ha comprobado que en dichos ejemplos una gran parte de la zona de búsqueda queda descartada debido a que en todos los puntos de esa zona se incumple la restricción sobre α .

Todo esto nos lleva a afirmar que OPCenum cumple su objetivo, que era proporcionar una manera fiable y eficiente de resolver el problema OPC-sb (4.7), resultando ser, además, una alternativa específica más eficaz que la aplicación de programas generales de optimización tales como los probados en la experiencia numérica del capítulo 3.

5.7. Conclusiones

En este capítulo hemos desarrollado un algoritmo específico para la resolución de problemas de precontrol óptimo: OPCenum. En concreto, el método propuesto resuelve la denominada versión semiacotada del precontrol óptimo (4.7), en la que las variables k y t se presentan acotadas superiormente, sin que ello suponga un condicionante importante en la práctica.

A partir de las propiedades del problema estudiadas en el capítulo 4, se ha propuesto un algoritmo basado en una estrategia enumerativa

implícita, que a su vez se apoya en un método de mejora local del parámetro λ , el cual emplea técnicas de búsqueda de raíces sin uso de derivadas.

Se han descrito dichos algoritmos en pseudocódigo, y se han dado detalles de su implementación en C++. Posteriormente, se han mostrado los resultados de aplicar el algoritmo OPCenum sobre la misma batería de ejemplos que se empleó en la experiencia numérica del capítulo 3.

A partir del análisis de los resultados obtenidos, puede concluirse que OPCenum supone una herramienta eficaz y eficiente para la determinación óptima de los parámetros del precontrol, en los términos en que ésta se planteó en el capítulo 3, es decir, persiguiendo minimizar el tamaño muestral medio del procedimiento cuando el proceso está centrado, y atendiendo a unas limitaciones exigidas por el usuario en términos de la máxima tasa de falsa alarma aceptada y la máxima probabilidad de no detectar una cierta desviación del proceso considerada inadmisibles.

Además, a pesar de la relativa sencillez del problema de programación matemática en que se ha materializado la búsqueda del mejor plan de precontrol, hay que destacar que el algoritmo enumerativo propuesto consigue algo que no se había alcanzado con el *software* de propósito general probado en la experiencia numérica del capítulo 3: OPCenum garantiza que la solución devuelta es la óptima o que, en su caso, el problema a resolver es imposible. A esto cabe añadir que el procedimiento es transparente para el usuario: éste no necesita saber configurar ningún parámetro u opción del algoritmo para que resuelva correctamente el problema, a diferencia de los algoritmos probados en el referido capítulo 3.

Todo ello permite concluir que se ha logrado uno de los objetivos de este trabajo: diseñar e implementar un algoritmo específico para el problema del precontrol óptimo que sea eficaz y eficiente, utilizando para ello el conocimiento obtenido al estudiar las propiedades de dicho problema.

Otras contribuciones y producción científica asociada

Más allá de los objetivos establecidos para la presente memoria, los cuales han sido alcanzados en los capítulos anteriores, creemos necesario completar a continuación la visión de todo el trabajo realizado en torno al precontrol en el marco de esta tesis. En primer lugar, se presenta un resumen de otras contribuciones propias realizadas en relación a dicha técnica; posteriormente se detalla toda la producción científica asociada.

6.1. Otras aportaciones al precontrol

Detallamos en esta sección otras contribuciones originales propias en torno al precontrol, fruto de trabajos en curso que han sido presentados ya en los foros adecuados (véase la sección 6.2). Se hará especial hincapié en el estudio preliminar de la extensión del precontrol óptimo a otras distribuciones de probabilidad y a otros ámbitos, como el de la fiabilidad (subsección 6.1.3).

6.1.1. Análisis de sensibilidad ante la ausencia de normalidad

A pesar de que la técnica clásica del precontrol es presentada como una metodología que no depende de la distribución de probabilidad de

la característica de calidad objeto de interés, lo cierto es que todos los enfoques, como el nuestro, que persiguen acotar las probabilidades para los errores tipo I y tipo II del precontrol necesitan presuponer una cierta distribución subyacente; habitualmente, la distribución normal.

Una pregunta a plantearse es: ¿cómo afecta la falta de normalidad a planes de precontrol que hayan sido diseñados bajo el supuesto de que la característica de calidad se distribuye según esa distribución? En concreto, pensando en el precontrol óptimo que hemos desarrollado: ¿cómo se verán modificados los valores del tamaño muestral esperado (función objetivo del problema OPC (4.5)), la tasa de falsa alarma (lado izquierdo de la primera restricción) y la probabilidad de no detectar una determinada desviación del proceso considerada inaceptable (segunda restricción)?

Resultados preliminares suponiendo diversas distribuciones subyacentes (en concreto, una distribución Weibull con la misma media y desviación típica que la normal supuesta inicialmente, y una Weibull con la misma proporción de piezas no conformes que ésta) fueron presentados por Giner y San Matías (2004b) y San Matías y Giner (2007)¹. Se concluyó que 1.º) en caso de detectar que el tamaño medio del muestreo obtenido es sensiblemente menor que el óptimo teórico, junto a un elevado porcentaje de cualificaciones, puede que el proceso presente una desviación de la distribución hacia una Weibull con menor dispersión que la asumida, y entonces el plan de precontrol puede ser excesivamente riguroso para los requerimientos dados; y 2.º) si el tamaño medio es menor que el teórico pero no aumenta excesivamente el porcentaje de cualificaciones, es posible que haya *asimetrización* de la distribución, con mayor tasa de falsa alarma que la deseada y produciendo más defectuosas, lo cual debería hacer reconsiderarse el plan de precontrol.

Los próximos pasos a dar para avanzar en esta línea de investigación se detallan en la subsección 7.2.5.

¹Todas las citas bibliográficas de este capítulo se encuentran referenciadas en la sección 6.2.

6.1.2. Enfoques no paramétricos

Una alternativa para salvar aquellos casos en que no se conozca con exactitud la distribución de probabilidad de la variable a estudiar es la aplicación de un enfoque no paramétrico, es decir, que no necesite presuponer ninguna distribución previa en los datos.

En este sentido, hemos explorado (Giner y San Matías, 2004a) el uso de la llamada verosimilitud empírica (*empirical likelihood*; Owen, 2001), un método que, a partir de los datos observados, intenta estimar una distribución de los mismos.

También se ha estudiado (Giner et al., 2005, 2006a, 2006b) el uso de redes neuronales de tipo perceptrón multicapa (Bishop, 1995), a veces identificado como un modelo de regresión *altamente no lineal*.

Posteriormente estableceremos los pasos a seguir en esta línea de investigación (subsección 7.2.6).

6.1.3. Aplicación en el ámbito de la fiabilidad

Presentamos a continuación en esta memoria una de las líneas de aplicación del precontrol óptimo en la que ya hemos estado trabajando.

Uno de los contextos donde surgen con más frecuencia las distribuciones asimétricas es en el ámbito de la fiabilidad y el mantenimiento de procesos o sistemas². El tiempo entre dos fallos consecutivos puede ser objeto de monitorización mediante herramientas de control de la calidad. Por ejemplo, los llamados *gráficos exponenciales* (Chan et al., 2000; Xie et al., 2002; Zhang et al., 2005; véase la subsección 1.2.2) se usan para monitorizar el tiempo entre sucesos —tiempo entre fallos, en un contexto de fiabilidad— en un proceso de Poisson homogéneo. Cada punto dibujado en el gráfico corresponde a una observación del

²La fiabilidad puede ser vista como una rama de la ingeniería de sistemas que trata del estudio de la aptitud de sistemas o componentes para funcionar correctamente a lo largo del tiempo, cuando operan en un entorno específico. El mantenimiento de sistemas es el conjunto de técnicas que aseguran el correcto funcionamiento de equipos, maquinaria, instalaciones y servicios, con el fin último de incrementar su fiabilidad. Puede consultarse, por ejemplo, el texto de Ebeling (1997) para más información.

tiempo transcurrido entre dos ocurrencias consecutivas, y los límites de control son fijados asumiendo que el proceso sigue una distribución exponencial. Los gráficos exponenciales y sus variantes han demostrado ser una herramienta válida para monitorizar la fiabilidad de sistemas (Di Bucchianico et al., 2005).

Una posible alternativa consistiría en adaptar el precontrol a la monitorización del tiempo entre fallos, suponiendo que éste sigue una distribución exponencial (véase también la sección 7.2.7). De hecho, es una propuesta en la que ya hemos comenzado a trabajar (San Matías y Giner-Bosch, 2008; Giner-Bosch y San Matías, 2009; San Matías y Giner-Bosch, 2009; Giner-Bosch y San Matías, 2010c; San Matías y Giner-Bosch, 2011). De manera breve presentamos a continuación el estado actual de desarrollo de esta línea de investigación.

Adaptación del modelo del precontrol óptimo al caso asimétrico

En este apartado presentamos la adaptación del precontrol óptimo al caso en el que la característica que está siendo observada sigue una distribución exponencial. Puede aplicarse al seguimiento del tiempo entre sucesos y, en particular, al tiempo entre fallos, en un contexto de fiabilidad.

Método de clasificación Supondremos que estamos monitorizando una variable aleatoria T con una distribución exponencial de media $\mu > 0$. Esto significa que su función de distribución acumulada puede expresarse de la forma siguiente:

$$F_T(t) = \Pr(T \leq t) = 1 - e^{-t/\mu}, t \geq 0. \quad (6.1)$$

El parámetro μ representa el tiempo medio entre dos fallos consecutivos (equivalentemente, la tasa media de fallos por unidad de tiempo sería $1/\mu$). Se asume que existe un valor deseado o inicialmente supuesto para dicho parámetro, al que llamaremos μ_0 .

Para adaptar el precontrol óptimo al caso exponencial, en primer lugar debe ajustarse la definición de las zonas verde, amarilla y roja, y poder así

clasificar las observaciones que se vayan obteniendo. Al estar trabajando con una distribución muy asimétrica, no tiene sentido seguir definiendo los límites para dichas zonas simétricos respecto de μ_0 . Nuestra propuesta actual para este tipo de distribuciones es fijar los límites de la zona verde G_L y G_U de manera que, cuando el tiempo medio es realmente μ_0 , las dos *subzonas* resultantes de dividir dicha zona por $\mu_0 \ln 2$ (la *mediana* de la variable T) sean igualmente probables, y lo mismo para las dos subzonas amarillas y las dos rojas. Esta idea está inspirada en cómo se definen en Chan et al. (2000) los límites de control para los gráficos exponenciales y, en definitiva, es la manera habitual de proceder cuando se construye un gráfico de control para una variable muy asimétrica y que, por tanto, no puede ser aproximada mediante una distribución normal.

En primer lugar nos referiremos a las tolerancias. Como sabemos, LSL y USL representan los límites del rango de valores admisibles para T . Es posible que en un contexto de fiabilidad no existan valores predeterminados para dichos límites. En tal caso, pueden fijarse de la siguiente forma:

$$LSL = -\mu_0 \ln(1 - p/2) \quad (6.2)$$

$$USL = -\mu_0 \ln(p/2) \quad (6.3)$$

donde p representa la probabilidad de que T esté fuera del rango de valores admisibles cuando el tiempo medio entre fallos es μ_0 .

En particular, una manera de determinar LSL y USL es fijando un valor para la capacidad potencial C_p . Dado que, por tratarse T de una variable exponencial, su desviación típica σ_T coincide con su media, podemos calcular el parámetro p a partir del valor asignado a C_p de la siguiente forma:

$$p = \frac{2e^{-6C_p}}{1 + e^{-6C_p}}. \quad (6.4)$$

Valores habituales para C_p , tales como 1, 1,2 o 1,5, no necesariamente son de aplicación aquí, puesto que surgen en procesos potencialmente capaces normalmente distribuidos, que no es el caso actual. Sin embargo,

la conclusión final sigue siendo la misma: cuanto mayor sea C_p , mayor es el rango de valores aceptables para T . Por otro lado, valores razonables para p , como 0,05, 0,1 o 0,2, pueden ser directamente asumidos.

Una vez los valores para LSL y USL han sido fijados, los límites de la zona verde pueden ser determinados mediante las siguientes expresiones:

$$G_L = \mu_0 \ln \left(1 + e^{-W_G/\mu_0} \right) \tag{6.5}$$

$$G_U = \mu_0 \ln \left(1 + e^{W_G/\mu_0} \right) \tag{6.6}$$

Estos límites se obtienen directamente teniendo en cuenta la condición previamente enunciada de que la probabilidad de que T tome valores inferiores a G_L coincida con la de que tome valores superiores a G_U , cuando el tiempo medio entre fallos es μ_0 .

La figura 6.1 muestra el aspecto de las zonas verde, amarilla y roja, cuando son definidas de esta manera.

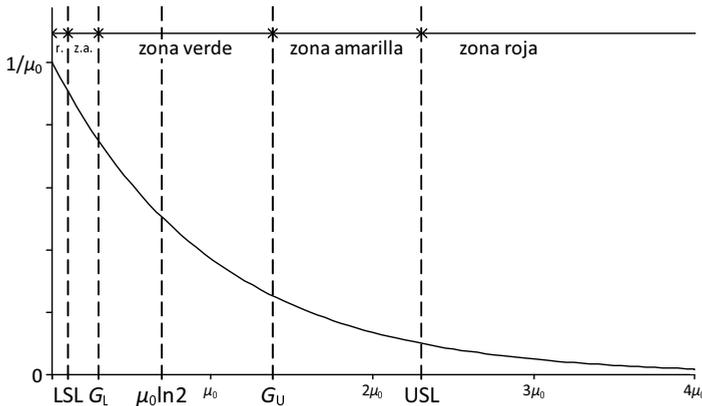


Figura 6.1: Definición de las zonas verde, amarilla y roja para un proceso exponencial.

Modelo de optimización Una vez hemos adaptado el método de clasificación del precontrol al caso exponencial, podemos desarrollar las expresiones para $p_g(\mu)$, $p_y(\mu)$ y $p_r(\mu)$, las probabilidades de obtener una

observación verde, amarilla o roja, respectivamente, cuando la media de T es μ . A partir de las expresiones (6.2), (6.3), (6.4), (6.5) y (6.6), obtenemos lo siguiente:

$$p_g(\mu) = \left(1 + e^{-W_G/\mu_0}\right)^{-\frac{\mu_0}{\mu}} - \left(1 + e^{W_G/\mu_0}\right)^{-\frac{\mu_0}{\mu}} \quad (6.7)$$

$$p_y(\mu) = \left(\frac{1}{1 + e^{-6C_p}}\right)^{\frac{\mu_0}{\mu}} - \left(\frac{1}{1 + e^{6C_p}}\right)^{\frac{\mu_0}{\mu}} - p_g(\mu) \quad (6.8)$$

$$p_r(\mu) = 1 - p_y(\mu) - p_g(\mu) \quad (6.9)$$

Para el caso particular de $\mu = \mu_0$, se tiene:

$$p_g(\mu_0) = \frac{1 - e^{-W_G/\mu_0}}{1 + e^{-W_G/\mu_0}} \quad (6.10)$$

$$p_y(\mu_0) = \frac{1 - e^{-6C_p}}{1 + e^{-6C_p}} - p_g(\mu_0) \quad (6.11)$$

Estas expresiones, particulares del caso exponencial, pueden incorporarse fácilmente a las representaciones generales de $E[n(\mu_0)]$, α y β_μ obtenidas en el capítulo 3 de esta memoria. El modelo general (3.1) quedaría adaptado de esta forma al caso exponencial:

$$\left. \begin{array}{l} \text{Min } E[n(\mu_0)] \\ \text{s.a: } \alpha \leq \alpha^* \\ \beta_{\mu_1^*} \leq \beta_1^* \\ \beta_{\mu_2^*} \leq \beta_2^* \\ \lambda \geq 2 \\ k, t \geq 1 \\ k, t \text{ enteras} \end{array} \right\} \quad (6.12)$$

Nótese que hemos incluido dos restricciones con el objeto de asegurar suficiente potencia para detectar desviaciones de la media respecto de μ_0 en ambas direcciones, ya que la variable que estamos monitorizando no es simétrica. Así, los parámetros μ_1^* y μ_2^* representan sendos valores para

el tiempo medio entre fallos que se desean detectar con probabilidades $1 - \beta_1^*$ y $1 - \beta_2^*$, respectivamente. Se supone que $\mu_1^* < \mu_0 < \mu_2^*$.

Nótese también que μ_2^* representa, por tanto, un incremento en el tiempo medio entre fallos —es decir, una *mejora* del proceso— que querríamos ser capaces de detectar.

Resolviendo este modelo de MINLP mediante técnicas de Optimización adecuadas, obtendríamos los valores de k , t y λ para configurar el procedimiento de Precontrol.

Un enfoque unilateral sería también posible, en el caso de que no se considere de interés la detección de mejoras del proceso. Las expresiones que hemos construido se podrían adaptar fácilmente, y el modelo de MINLP que se obtendría sería similar al de la expresión (6.12), excepto en que la restricción $\beta_{\mu_2^*} \leq \beta_2^*$ no estaría presente.

Ventajas

A continuación resaltamos las ventajas que ofrece esta adaptación del precontrol, como método para monitorizar la fiabilidad de un sistema.

En primer lugar, nuestro enfoque mantiene las ventajas de los gráficos exponenciales frente a los gráficos de control tradicionales, como son:

- No requiere determinar un tamaño muestral para realizar subgrupos de observaciones, como sí ocurre en los gráficos de control tradicionales, ya que cada observación es *graficada* o tratada directamente.
- Su comportamiento es bueno en procesos con alto rendimiento, es decir, sistemas con una tasa de fallo muy baja, en un contexto de fiabilidad.
- Permite detectar no sólo el deterioro sino también la mejora del proceso, esto es, descensos en la tasa media de fallos.

A esto hay que añadir una ventaja particular de nuestra técnica, y que la distingue de los gráficos exponenciales. El precontrol, como ya sabemos, *no precisa dibujar* los datos; únicamente necesitamos llevar un

conteo de los fallos *verdes, amarillos y rojos* que se van produciendo. Y para saber si un fallo observado es de uno u otro tipo, *no es necesario monitorizar de manera continua el sistema*, sino únicamente comprobar si ha habido algún fallo en los instantes indicados por el plan de precontrol obtenido, es decir, en los instantes LSL , G_L , G_U y USL , a contar a partir de la puesta en marcha del sistema tras el último fallo registrado. Esto se pone de manifiesto en el algoritmo esquematizado que presentamos a continuación.

Implementación

Presentamos una posible forma de implementar el precontrol para monitorizar tiempos de fallo. Ha sido desarrollada para el planteamiento bilateral que hemos asumido en toda esta sección, pero sería fácilmente adaptable al caso unilateral.

FASE DE DISEÑO

- PASO 1. Estimar/determinar μ_0 (el tiempo medio entre fallos actual/deseado), y determinar LSL y USL (o, equivalentemente, p o C_p).
- PASO 2. Determinar las condiciones iniciales α^* , μ_1^* , μ_2^* , β_1^* y β_2^* .
- PASO 3. Encontrar los valores óptimos para los parámetros del precontrol k , t , λ , resolviendo el modelo (6.12). Calcular G_L y G_U usando las expresiones (6.5) y (6.6), respectivamente.

FASE DE EJECUCIÓN

- PASO 1. Comenzar un nuevo gráfico (lo cual significa, sencillamente, inicializar el conteo de verdes, amarillos y rojos a cero). Inicializar el tiempo de observación e iniciar el sistema o proceso cuya fiabilidad queremos monitorizar.
- PASO 2. Comprobar si se ha producido algún (nuevo) fallo al menos en los instantes determinados por el precontrol, es decir, cuando el tiempo desde el último arranque del sistema es LSL , G_L , G_U y USL .

- PASO 3. Graficar/contar un *verde* si un fallo ha tenido lugar entre los instantes G_L y G_U , un *amarillo* si un fallo ha ocurrido dentro del intervalo de tiempo $[LSL, G_L[$ o $]G_U, USL]$, o un *rojo* si un fallo ha sucedido antes del instante LSL o si no se ha observado ningún fallo hasta el instante USL.
- PASO 4. Si k fallos verdes consecutivos se han observado, entonces el proceso está cualificado, es decir, puede asumirse un valor de $1/\mu_0$ para la tasa de fallos del sistema. Detener la monitorización hasta la próxima comprobación programada, y volver entonces a la FASE DE EJECUCIÓN.
- PASO 5. En otro caso, si t fallos amarillos consecutivos o 1 fallo rojo han sido observados, entonces dos situaciones diferentes pueden darse:
 - Si el tiempo medio entre fallos observado es menor que μ_0 , el sistema puede haberse deteriorado. En tal caso, deben tomarse las acciones pertinentes para identificar y eliminar el problema.
 - Si el tiempo medio entre fallos observado es mayor que μ_0 , el sistema puede haber experimentado una mejora. Las causas de tal supuesta mejora deberían detectarse y, de ser posible, consolidarse.

Tras ello, si μ_0 o las condiciones iniciales han cambiado, ir a la FASE DE DISEÑO. Si no, reiniciar la FASE DE EJECUCIÓN.

- PASO 6. En otro caso, reparar y reiniciar el sistema si es necesario, y reinicializar el tiempo de observación. Volver al PASO 2.

Ejemplos numéricos

San Matías y Giner-Bosch (2008) presentan un sencillo ejemplo numérico, el cual reproducimos a continuación, para ilustrar de qué manera el precontrol óptimo puede ser aplicado en la monitorización de

la fiabilidad de un sistema, de acuerdo a la implementación que acabamos de proponer.

Supongamos que se tiene un sistema que presenta un tiempo entre fallos distribuido de modo exponencial, con un tiempo medio entre fallos inicialmente asumido de $\mu_0 = 1000$ horas.

Se ha elegido un valor de $p = 0,0027$ ($C_p = 1,10$) para determinar los límites de especificaciones $LSL = 1,35$ y $USL = 6607,65$ horas. Además, la cota superior para la tasa de falsa alarma es fijada a $\alpha^* = 0,05$, y las probabilidades de no detectar un cambio en el tiempo medio entre fallos de $\mu_1^* = 100$ y $\mu_2^* = 5000$ horas son limitados superiormente por $\beta_1^* = \beta_2^* = 0,1$.

Con estas condiciones, el valor óptimo de los parámetros del precontrol de acuerdo con el modelo (6.12) son $k = 3$, $t = 4$ y $\lambda = 6,8233$. Por tanto, $W_G = 1936,38$, según la ecuación (2.24). Esto significa que los límites para la zona verde óptima son $G_L = 134,73$ y $G_U = 2071,11$ horas (usando las expresiones (6.5) y (6.6)), y que tenemos que observar 3 fallos verdes consecutivos para cualificar el sistema, y 4 fallos amarillos consecutivos o uno rojo para detenerlo y revisarlo. La solución del modelo (6.12) nos informa, además, de que el número esperado de fallos que tendremos que observar para terminar el procedimiento de precontrol, en el caso de que no haya ninguna desviación en el tiempo medio de fallos, es $E[n(\mu_0)] = 5,35$.

Se han simulado datos de un sistema como el supuesto, para llevar a cabo este ejemplo. La tabla 6.1 contiene una muestra de 45 tiempos de fallo simulados con una media de $\mu = \mu_0 = 1000$ horas, mientras que la tabla 6.2 corresponde a datos simulados con un tiempo medio de $\mu = \mu_1^* = 100$ horas.

La figura 6.2 muestra un gráfico para los datos de la tabla 6.1, e ilustra cómo el esquema de precontrol termina cualificando un sistema correcto. En el total de la simulación llevada a cabo con $\mu = \mu_0 = 1000$ horas, la tasa observada de falsa alarma fue 1,54% (ese fue el porcentaje de veces en que se produjo una alarma por deterioro o mejora del sistema en nuestra simulación), y el tamaño muestral medio necesario para una

1 a 9	10 a 18	19 a 27	28 a 36	37 a 45
1597,84	980,642	729,937	2934,51	702,022
58,8161	127,234	692,515	372,409	349,842
754,429	1163,61	25,2186	1374,91	1485,59
101,445	802,918	1437,56	148,811	1630,53
1206,92	717,654	471,385	417,410	3531,43

Tabla 6.1: Primeros 45 valores simulados para $\mu = \mu_0 = 1000$ horas en la experiencia numérica de la subsección 6.1.3.

1 a 9	10 a 18	19 a 27	28 a 36	37 a 45
17,85	22,9	46,93	29,58	69,44
68,4	174,11	46,11	18,9	71,81
13,15	231,7	11,73	77,22	84,54
83,92	23,57	60,19	48,03	111,15
76,88	39,97	239,64	3,24	206,9

Tabla 6.2: Primeros 45 valores simulados para $\mu = \mu_1^* = 100$ horas en la experiencia numérica de la subsección 6.1.3.

ejecución fue de 5,10 fallos.

La figura 6.3 corresponde a los datos de la tabla 6.2. En ella se muestra la aparición de alarmas por deterioro del sistema cuando éste presenta un tiempo medio entre fallos de $\mu = \mu_1^* = 100$ horas. El porcentaje de veces que se observó tal alarma en el total de la simulación fue 89,58% (un 13% de ellos debidos a un *fallo rojo*), lo cual significa que en un 10,42% de las veces el procedimiento del precontrol terminó cualificando el sistema por observar 3 verdes consecutivas (dicho porcentaje es similar a la probabilidad $\beta_1^* = 0,1$, establecida previamente, de no detectar un deterioro de esa magnitud en el sistema). El número medio de fallos que fue necesario observar para terminar cada ejecución fue 6,89.

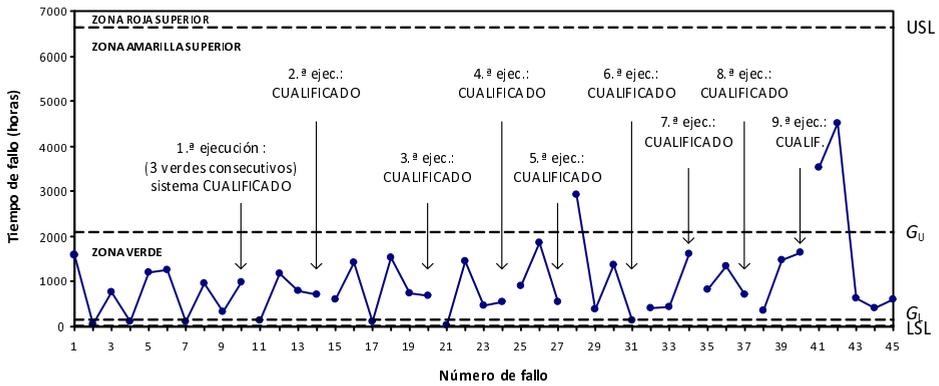


Figura 6.2: Gráfico de precontrol para la simulación de un sistema con un tiempo medio entre fallos de $\mu_0 = 1000$ horas (véase la tabla 6.1).

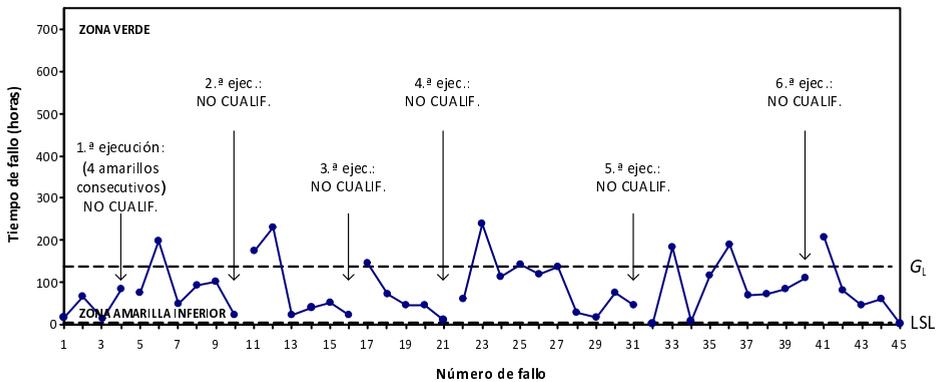


Figura 6.3: Gráfico de precontrol para la simulación de un sistema con un tiempo medio entre fallos de $\mu_1^* = 100$ horas (véase la tabla 6.2).

Conclusiones

El ejemplo anterior muestra la utilidad de nuestra propuesta para monitorizar el tiempo entre fallos, en un contexto de fiabilidad.

En la subsección 7.2.8 sugerimos los posibles caminos por los que profundizar en esta interesante línea de aplicación del precontrol óptimo.

6.1.4. Otras estrategias de resolución

A partir de todo lo expuesto en los capítulos 4 y 5, es claro que la estrategia enumerativa que hemos aplicado resuelve de manera suficientemente eficiente el problema OPC. Sin embargo, tal y como se mencionó en la introducción del capítulo 5, el algoritmo propuesto se obtuvo tras explorar diferentes técnicas de búsqueda de soluciones, de manera simultánea al estudio teórico que finalmente dio lugar al capítulo 4. En concreto, se realizaron pruebas con diversas técnicas heurísticas (véase la subsección 1.4.2), y en particular con la conocida como *scatter search* (búsqueda dispersa, en español), uno de los procedimientos metaheurísticos basados en metáforas evolutivas con mayor relevancia en las últimas décadas. Estas pruebas, con resultados interesantes, fueron presentadas en diversos congresos (Giner-Bosch y San Matías, 2010a; San Matías y Giner-Bosch, 2010b).

En concreto, las experiencias numéricas llevadas a cabo mostraron que la búsqueda dispersa diseñada logra alcanzar, con un esfuerzo computacional razonable, las soluciones óptimas obtenidas en el capítulo 5 para la batería de problemas considerada a lo largo de esta tesis.

En la subsección 7.2.9 destacamos las ventajas de la búsqueda dispersa, y exponemos en qué sentido pensamos que esta técnica puede ser útil en relación al precontrol óptimo.

6.2. Contribuciones en el ámbito científico

La labor realizada y expuesta en esta memoria supone la culminación de una etapa en la línea de investigación que el autor inició bajo la supervisión de la profesora D.^a Susana San Matías Izquierdo en el contexto del programa de doctorado dentro del cual se encuadra la presente tesis, y que tuvo como uno de sus primeros hitos la presentación en el año 2007 del trabajo de investigación «*Generación de planes de precontrol óptimos mediante técnicas de programación no lineal entera mixta*» (véase la subsección 6.2.6), el cual dio, al mismo tiempo, acceso a la consecución del Diploma de

Estudios Avanzados.

Asimismo, un avance del trabajo que comprende esta memoria fue publicado como *technical report* en el año 2010 con el título de «*Selection of best pre-control technique by optimization tools*» (véase el detalle en la subsección 6.2.5).

Un artículo de investigación basado en los resultados obtenidos en esta tesis doctoral ha sido recientemente enviado a la revista *Quality and Reliability Engineering International*, indexada en Web of Science y con índice de impacto de 0,994 en 2013 según el Journal Citation Reports (lo que la sitúa en el segundo tercio del *ranking* en las categorías de Ingeniería Industrial, Ingeniería Multidisciplinar e Investigación Operativa); se encuentra en proceso de revisión en el momento de redactar estas líneas.

Además de lo anterior, esta labor de investigación se ha materializado en forma de comunicaciones a congresos científicos. Todo ello se detalla a continuación, ordenado, en el caso de los congresos, por orden cronológico y agrupado según el alcance (nacional o internacional).

En la mayoría de estas contribuciones se tratan o exploran, de manera preliminar en algunos casos, aspectos del precontrol óptimo que han sido incluidos en la sección 6.1 de la presente memoria. Dichos aspectos son objeto asimismo de algunas de las futuras líneas de investigación que se introducen en el capítulo siguiente.

6.2.1. Publicaciones en revistas científicas indexadas

Giner-Bosch V, San Matías S, Clemente-Císcar M, Carrión A (2014). Optimal design of pre-control plans. *Quality and Reliability Engineering International*, bajo revisión.

6.2.2. Publicaciones en congresos internacionales indexados

San Matías S, **Giner-Bosch V** (2008). Optimal Pre-control as a tool to monitor the reliability of a manufacturing system. En: Martorell S, Soares CG, Barnett J (eds.), *Safety, reliability and risk analysis: Theory, methods and applications. European Safety and Reliability Conference / 17th Annual Meeting of the Society for Risk Analysis - Europe (ESREL 2008 & 17th SRA-Europe)*, volumen 4, págs.

2735–2741. CRC Press, València (España). ISBN: 978-0-415-48513-5. Indexada en Web of Scince.

6.2.3. Comunicaciones a congresos internacionales

Giner V, San Matías S (2003a). A methodology for the optimal determination of parameters in quality control tools. *New opportunities for Operations Research. 5th EURO/INFORMS Joint International Meeting (EURO/INFORMS 2003)*, pág. 109. Boğaziçi Üniversitesi, Estambul (Turquía).

Edo MÁ, **Giner V**, San Matías S (2004). Numerical experiments with Mixed Integer Nonlinear Programming techniques applied to quality control tools. *12th French-German-Spanish Conference on Optimization (FGS-2004)*. Université d'Avignon, Aviñón (Francia).

Giner V, San Matías S (2004a). Use of empirical likelihood in optimal pre-control plans. *Veszprém Optimization Conference: Advanced Algorithms (VOCAL 2004)*. Faculty of Information Technology, University of Veszprém, Veszprém (Hungría).

Giner V, San Matías S, Calduch MÁ, Edo MÁ (2005). An artificial neural network approach to pre-control. *17th EURO Mini Conference: Continuous Optimization in Industry*, pág. 39 University of Pécs, Pécs (Hungría).

Giner V, Calduch Á, San Matías S (2006a). A comparison between optimal and neural pre-control plans. *17th Conference of IASC-ERS (COMPSTAT 2006)*. Università degli Studi di Roma La Sapienza, Roma (Italia).

San Matías S, **Giner V** (2007). Behavior of optimal pre-control plans under non-normality. A numerical analysis. *OR creates bridges. 22nd European Conference on Operational Research (EURO 2007)*, pág. 171. University of Economics, Prague, Praga (República Checa).

Giner-Bosch V, San Matías S (2010a). Heuristic strategies for optimal design of quality control tools. *24th European Conference on Operational Research (EURO XXIV)*, pág. 129. Universidade de Lisboa, Lisboa (Portugal).

Giner-Bosch V, San Matías S (2010b). Optimal design of quality control tools via MINLP. Our experience with pre-control. 19th *International Conference on Computational Statistics (COMPSTAT 2010)*, pág. 404. Conservatoire National des Arts et Métiers/French National Institute for Research in Computer Science and Control, París (Francia).

Giner-Bosch V, San Matías S (2010c). Monitoring time between events in an exponentially distributed process by using optimal pre-control. *Exploratory Workshop on Mixed Integer Non-Linear Programming: Theory, Algorithms and Applications (EWMINLP, 1st edition)*. Instituto de Matemáticas Universidad de Sevilla, Universidad de Sevilla, Sevilla (España).

San Matías S, **Giner-Bosch V** (2011). Quality control tools for the monitoring of systems reliability. *International Conference on Dependability and Quality Management (ICDQM-2011, 2nd edition)*. The DQM Research Center, Belgrado (Serbia). ISBN: 978-86-86355-06-5.

6.2.4. Comunicaciones a congresos nacionales

Giner V, San Matías S (2003b). Una metodología para la determinación óptima de parámetros en el Precontrol. 27.^o *Congreso Nacional de Estadística e Investigación Operativa (SEIO 2003)*, págs. 873–878. Edicions de la Universitat de Lleida, Lleida. ISBN: 978-84-8409-955-0.

Giner V, San Matías S (2004b). Análisis de sensibilidad de los planes de precontrol óptimos frente a la no normalidad. *XXVIII Congreso Nacional de Estadística e Investigación Operativa (SEIO 2004)*, págs. 697–698. Departamento de Estadística e Investigación Operativa, Universidad de Cádiz, Cádiz. ISBN: 978-84-689-0438-2.

Giner V, San Matías S, Calduch Á (2006b). Una alternativa al precontrol clásico basada en redes neuronales artificiales. *XXIX Congreso Nacional de Estadística e Investigación Operativa (SEIO 2006)*, págs. 333–334. Departamento de Estadística, Investigación Operativa y Computación, Universidad de La Laguna, Puerto de la Cruz, Tenerife. ISBN: 978-84-689-8553-4.

Giner-Bosch V, San Matías S (2007). Evaluación del precontrol de calidad y sus recientes alternativas. Discusión y experiencias numéricas. *XXX Congreso*

Nacional de Estadística e Investigación Operativa (SEIO 2007). Valladolid. ISBN: 978-84-690-7249-3.

Giner-Bosch V, San Matías S (2009). Monitorización del tiempo entre fallos de un sistema mediante precontrol óptimo. *XXXI Congreso Nacional de Estadística e Investigación Operativa (SEIO 2009)*. Departamento de Estadística e Investigación Operativa, Universidad de Murcia, Murcia. ISBN: 978-84-691-8159-1.

San Matías S, **Giner-Bosch V** (2009). Monitorización del tiempo entre fallos. *Segundo Foro de Control Estadístico de la Calidad y Fiabilidad (CEC 2009)*. Universidade da Coruña, Ferrol.

San Matías S, **Giner-Bosch V** (2010b). Heurísticos para el precontrol óptimo. *XXXII Congreso Nacional de Estadística e Investigación Operativa (SEIO 2010)*. Universidade da Coruña, La Coruña. ISBN: 978-84-693-6152-8.

6.2.5. *Technical reports*

San Matías S, **Giner-Bosch V** (2010a). *Selection of Best Pre-Control Technique by Optimization Tools*. Technical report DEIOAC-2010-06. Departamento de Estadística e Investigación Operativa Aplicadas y Calidad, Universitat Politècnica de València, València. http://www.upv.es/deioac/Investigacion/prec_optimo.pdf

6.2.6. *Trabajos académicos*

Giner i Bosch V (2007). *Generación de planes de precontrol óptimos mediante técnicas de programación no lineal entera mixta*. Trabajo de investigación del Doctorado. Tutora: Susana San Matías Izquierdo. Departamento de Estadística e Investigación Operativa Aplicadas y Calidad, Universitat Politècnica de València, València.

Conclusiones y líneas futuras de investigación

7.1. Conclusiones

El precontrol es una de las técnicas existentes para el control de la calidad especialmente orientadas a procesos con ejecuciones cortas y ciclos rápidos. Trata directamente con las especificaciones del producto. Su rapidez, sencillez y facilidad de uso hacen de ésta una técnica atractiva, si bien en su versión clásica presenta ciertos aspectos mejorables, relativos a la tasa de falsa alarma y la capacidad para detectar desviaciones inaceptables del proceso en determinadas circunstancias, debidos, entre otros motivos, a que la técnica no tiene en cuenta la variabilidad natural del proceso.

En el presente trabajo se ha estudiado la optimización de la fase de cualificación del precontrol como herramienta de control de la calidad. Partiendo de la mejora del precontrol propuesta por [San Matías et al. \(2004\)](#), consistente en flexibilizar el valor de los parámetros que rigen el comportamiento de la técnica, se ha aplicado un enfoque de optimización orientado a determinar la mejor combinación de dichos parámetros, en función de unos requerimientos dados, y con el fin último de desarrollar una metodología que permita automatizar dicha elección.

Para ello, en el **capítulo 2** se ha realizado, como primer paso, una revisión de las variantes y mejoras del precontrol encontradas en la literatura, partiendo de la técnica clásica y llegando hasta las propuestas aparecidas recientemente, convirtiéndose así en la revisión más completa y actualizada sobre el tema, hasta donde hemos podido conocer.

En el mismo capítulo se ha descrito con detalle el denominado precontrol generalizado, propuesto por [San Matías et al.](#) y ya antes mencionado. Se ha mostrado cómo esta alternativa supone un avance respecto al resto de mejoras existentes, en el sentido de que, gracias a la flexibilización que plantea, permite ajustarse a unos valores deseados para la tasa de falsa alarma y la potencia para detectar desviaciones inaceptables, salvando así las principales debilidades de la técnica clásica.

Por último, en la sección 2.5 de dicho capítulo se ha descrito de manera formal cómo evaluar el comportamiento de un plan de precontrol dado a través de su potencia para detectar desajustes del proceso, bien considerado como un contraste sobre la estabilidad de la media proceso, o bien como contraste sobre la capacidad del mismo, constituyendo este último enfoque una contribución propia original. Asimismo, en dicha sección se ha demostrado que la expresión obtenida por [San Matías et al.](#) para la probabilidad de cualificar un proceso está bien definida, en el sentido de que está definida para cualquier valor de los parámetros involucrados, y de que su valor nunca es inferior a 0 ni superior a 1. Puede consultarse asimismo la sección de conclusiones del capítulo 2 en la página 99 de esta memoria.

En el **capítulo 3** hemos desarrollado nuestra propuesta, consistente en plantear la determinación óptima de los parámetros del precontrol, tal y como son descritos por [San Matías et al.](#) Esto se ha concretado en la formulación de un modelo de programación no lineal entera mixta, la expresión concreta de cuya función objetivo se ha obtenido en el propio capítulo. Se ha realizado una completa experiencia numérica, apoyada en *software* de optimización de propósito general, que permite concluir que el modelo desarrollado es útil y se adecua al objetivo pretendido.

Asimismo, la experiencia numérica llevada a cabo pone de manifiesto

que el uso de *software* general de MINLP para la obtención de planes de precontrol óptimos por parte de usuarios no avanzados puede conllevar ciertas dificultades, en parte intrínsecas a la propia naturaleza del problema. De todos los algoritmos evaluados, aquel que ofreció un mejor comportamiento fue GAMS/DICOPT. Una relación detallada de las conclusiones de dicha experiencia puede consultarse en la página 193 de esta tesis. Igualmente, las conclusiones globales del capítulo 3 se encuentran en la página 199.

En el **capítulo 4** se ha realizado un estudio teórico de las propiedades de las funciones involucradas en el precontrol óptimo. Este estudio prueba matemáticamente diversas características del modelo propuesto en el capítulo 3, la mayoría de las cuales, pese a ser intuitivamente ciertas, no poseen una demostración trivial, como se ha visto.

En concreto, se ha demostrado que la probabilidad de cualificar el proceso es estrictamente decreciente en relación al parámetro del precontrol λ (inversamente proporcional a la anchura de la zona verde), fijadas el resto de condiciones, y que el tamaño muestral esperado para terminar el procedimiento de cualificación posee, de nuevo como función de λ y fijados el resto de parámetros, un único máximo y ningún mínimo local.

También se han enunciado las condiciones mínimas que debe cumplir la característica de calidad objeto de interés para que le sean aplicables todos los resultados demostrados. La sección de conclusiones del capítulo se encuentra en la página 289.

Las propiedades demostradas en el capítulo 4, además de permitir profundizar en el conocimiento del precontrol, suponen el fundamento teórico para un algoritmo exacto específico para la resolución de problemas de precontrol óptimo, que hemos desarrollado en el **capítulo 5**, basado en una estrategia de búsqueda enumerativa implícita, empleando auxiliarmente métodos de búsqueda de raíces sin uso de derivadas. Este algoritmo representa, junto con la elaboración del modelo del precontrol óptimo, la principal contribución de la presente tesis doctoral.

Se ha proporcionado el pseudocódigo del algoritmo y se ha detallado

su implementación en C++. Las experiencias numéricas llevadas a cabo y descritas en el mismo capítulo confirman la eficacia y eficiencia de nuestra propuesta, la cual, además, produce resultados más fiables que los derivados del uso de algoritmos de optimización de propósito general. Pueden consultarse las conclusiones del capítulo 5 en la página 335 de esta memoria.

Hay que hacer notar que, en trabajos similares del ámbito de la optimización en control de la calidad, los algoritmos específicos que se proponen para los problemas objeto de interés son habitualmente de tipo heurístico (algoritmos genéticos, principalmente), sin que ni siquiera suela considerarse la posibilidad de resolverlos de manera exacta. Se desprende de ello la novedad y la relevancia del enfoque exacto con que se ha abordado el problema del precontrol óptimo.

Por último, en el capítulo 6 se ha presentado de manera resumida, entre otros trabajos en curso, una investigación preliminar de la adaptación del precontrol óptimo al caso asimétrico y su aplicación al ámbito de la fiabilidad de sistemas. Los resultados hacen posible afirmar que se trata de una línea de investigación prometedora.

Todo lo dicho permite resumir concluyendo que se han cubierto los objetivos del presente trabajo: *plantear formalmente la determinación de los parámetros del precontrol como problema de optimización y diseñar e implementar algoritmos específicos para dicho problema, además de estudiar de manera preliminar la extensión del precontrol óptimo a otras distribuciones de probabilidad y explorar su aplicación a otros ámbitos más allá del control de la calidad*. El trabajo realizado hace posible, además, pensar en otros enfoques, alternativas y aplicaciones, que serán detallados en la sección siguiente.

7.2. Líneas futuras de investigación

La investigación llevada a cabo y los resultados obtenidos acerca de la optimización del precontrol en el marco de la presente tesis doctoral sugieren la extensión de esta línea de investigación en varias

direcciones, algunas de las cuales se han ido avanzando a lo largo de la presente memoria; otras de ellas han sido ya objeto de estudios preliminares presentados en diversos congresos, como se ha mencionado en el capítulo 6. Todo ello se recoge a continuación.

7.2.1. Desarrollo de una interfaz gráfica

El algoritmo específico para el precontrol óptimo que hemos desarrollado en esta tesis doctoral permite alcanzar la solución óptima, en caso de haberla, en un tiempo aceptable. Dicho algoritmo se ha implementado como programa de consola en C++, como ya se mencionó en la subsección 5.5.1, el cual, tras ser compilado, puede ejecutarse sin necesidad de tener instalado el entorno Visual Studio ni ningún otro compilador o intérprete de C++. No obstante, con el fin de hacer nuestra propuesta más accesible y aplicable en un entorno industrial, nos planteamos dotar en el futuro al programa elaborado de una interfaz gráfica fácil de usar.

Dicha interfaz podría construirse directamente en el mismo entorno Visual Studio, o en cualquier otro entorno. La funcionalidad básica sería la de permitir al usuario obtener el plan de precontrol óptimo a partir de la introducción de los parámetros del problema.

Además de esto, otras características que nos planteamos añadir a la interfaz son las siguientes:

- En cuanto a la introducción de datos:
 - Permitir al usuario la posibilidad de introducir la información sobre la desviación inaceptable δ^* indirectamente, en términos de la máxima proporción admisible de piezas no conformes p^* (véase la expresión (2.34)).
 - Permitir al usuario la posibilidad de introducir la información sobre la capacidad del proceso C_p dando directamente los límites de tolerancias LSL y USL, y una estimación de la desviación típica cuando el proceso está bajo control σ_0 .

- En un un nivel de complejidad mayor, permitir la posibilidad de que dichos parámetros (δ^* y σ_0) sean estimados por el propio programa a partir de datos históricos facilitados por el usuario o accesibles desde alguna ubicación.
- En cuanto a la visualización de resultados:
 - Mostrar no sólo el valor de λ sino también, obviamente, el valor de la anchura de la zona verde en relación a la anchura del rango de tolerancias (tal y como se hizo en el ejemplo numérico de la subsección 5.4.5).
 - Mostrar información básica adicional, tal como el valor final de la tasa de falsa alarma α y la probabilidad β_{δ^*} de no detectar la desviación introducida como inaceptable (es decir, los respectivos lados izquierdos de las dos restricciones del problema OPC), para el plan de precontrol óptimo devuelto.
 - Mostrar gráficamente la curva de potencia y/o la curva característica en función de la desviación δ asociada al plan de precontrol óptimo.
 - Mostrar la mejor solución para diversas combinaciones de k y t , con el fin de que el usuario pueda elegir la que más le convenga.

Como alternativa a la implementación de la interfaz gráfica en C++, contemplamos la posibilidad de adaptar el algoritmo para su uso en dispositivos móviles con sistema operativo Android, iOS y Windows Phone, a fin de acercar lo máximo posible la herramienta al usuario, bien a través de aplicaciones nativas específicas, o bien mediante una interfaz web con diseño adaptativo.

Finalmente, otra vía a explorar muy interesante es la posibilidad de implementar el algoritmo del precontrol óptimo en Mathematica, aprovechando la potencia y versatilidad de dicha plataforma y las funcionalidades de programación que ofrece. Además, Mathematica

permite crear documentos no modificables por un tercero pero con cierto nivel de interactividad, pensados para ser abiertos con un visor externo y gratuito llamado *CDF¹ Player*. Las posibilidades de este formato, unidas a la facilidad y potencia de Mathematica para realizar y mostrar cálculos, gráficos interactivos, etc., permitirían poner al alcance del usuario final una implementación del algoritmo que hemos diseñado muy completa, útil, accesible y con un aspecto profesional.

7.2.2. Estudio del peor caso

En la subsección 3.5.3 se ha realizado un estudio descriptivo del peor caso, es decir, del máximo valor que puede alcanzar el tamaño muestral promedio de un plan de precontrol óptimo dado en función de la desviación real del proceso.

Sería interesante abordar el tema desde un punto de vista formal, de modo similar a como se ha hecho con el estudio de las expresiones p_Q y f como funciones de λ en la subsección 4.3.1. La principal dificultad reside en el hecho de que las probabilidades p_g y p_y no son monótonas respecto de la desviación δ , fijados el resto de parámetros (véase la figura 2.9), como sí lo eran como funciones de λ .

Alternativamente, podría estudiarse la búsqueda del máximo de f como función de δ desde un punto de vista numérico, aplicando técnicas de optimización no lineal.

En cualquier caso, esto permitiría tanto idear formas de modificar el modelo convenientemente como, sencillamente, poder dar una información precisa al usuario final acerca de cuál es la peor situación que podría producirse.

Por último, otra posibilidad, orientada directamente a modificar el modelo y ya mencionada en la subsección 3.3.3, es abordar el caso de minimizar el máximo tamaño muestral como un problema de programación semiinfinita.

¹Siglas de *Computable Document Format*, formato de documento computable. Podría describirse de manera informal como una especie de PDF interactivo.

7.2.3. Adecuación del modelo a otros requerimientos

En las secciones 3.2 y 3.3 se ha construido de manera razonada el que finalmente se ha considerado como modelo del precontrol óptimo. No obstante, podrían plantearse a partir de él diferentes modificaciones, como por ejemplo:

- añadir otras restricciones complementarias, encaminadas —entre otras posibles alternativas— a acotar inferiormente las respectivas potencias para detectar diferentes desviaciones inaceptables y/o la probabilidad de aceptar aquellas que sean consideradas despreciables;
- considerar límites de tolerancia y/o de precontrol asimétricos respecto del valor nominal (que serían de utilidad, por ejemplo, en el caso de que la distribución de probabilidad subyacente no fuese simétrica; véanse las subsecciones 7.2.7 y 7.2.8);
- considerar otras funciones objetivo (por ejemplo, el tamaño muestral esperado en el caso de producirse una cierta desviación inaceptable); o
- como consecuencia del estudio del peor caso (véase la subsección 7.2.2), añadir otras restricciones que se consideren convenientes para mejorar el modelo en este aspecto; etc.

Estas y otras modificaciones tendrían sentido en tanto en cuanto sean relevantes y proporcionen soluciones que sean razonables desde un punto de vista práctico.

7.2.4. Precontrol óptimo como contraste sobre la capacidad

Un enfoque paralelo al desarrollado en esta memoria es el de optimizar los parámetros del precontrol considerándolo como contraste sobre la capacidad del proceso C_p . En la subsección 2.5.2 ya se han mostrado las

ideas principales de esta alternativa, incluyendo curvas características para distintos valores de los parámetros de la técnica.

Se trata de una línea de investigación que, desde un punto de vista aplicado, sin duda merece la pena abordar, ya que en muchas ocasiones el precontrol y otras herramientas similares son utilizadas, precisamente, para valorar la capacidad de determinado un proceso.

Asimismo, inspirados en el trabajo de [Cox et al. \(2013b\)](#) (véase la subsección 2.2.8), nos planteamos trabajar directamente con la capacidad real C_{pk} (1.3), para, en la medida de lo posible, monitorizar a un tiempo la desviación y la variabilidad del proceso en relación a las tolerancias del mismo.

7.2.5. Análisis de sensibilidad ante la ausencia de normalidad

En la subsección 6.1.1 se han presentado las conclusiones de un estudio numérico preliminar llevado a cabo sobre las consecuencias de aplicar un plan de precontrol óptimo diseñado bajo la hipótesis de normalidad a procesos que siguen una distribución Weibull.

Dichas conclusiones deberían ser contrastadas y completadas con más experimentos que contemplen otras distribuciones de probabilidad, así como el estudio de la posible influencia de los parámetros del problema (C_p , α^* , β^* y δ^*) sobre las desviaciones observadas.

7.2.6. Enfoques no paramétricos

En la subsección 6.1.2 se han mencionado nuestros primeros intentos en la aplicación de técnicas no paramétricas como la verosimilitud empírica y redes neuronales tipo perceptrón multicapa, para la obtención de planes de precontrol óptimos en casos en que se desconozca la distribución subyacente en el proceso objeto de interés.

A pesar de la complejidad que comportan, son dos enfoques novedosos e interesantes que requieren todavía de más trabajo para poder obtener resultados concluyentes.

7.2.7. Adaptación a otras distribuciones de probabilidad

Tal y como se ha destacado en la introducción de la presente memoria (sección 1.5), las hipótesis mantenidas en el desarrollo de nuestra propuesta (capítulos 3 a 5), esto es, la normalidad de la característica de calidad objeto de interés y la simetría de los límites de precontrol y de las tolerancias respecto del valor objetivo, son bastante lógicas y habituales en control de la calidad.

Con todo, tal y como se ha expuesto en la sección 4.5, el enfoque aplicado es trasladable de manera sencilla a contextos en los que la característica de calidad objeto de interés se ajuste a otras distribuciones de probabilidad.

Ahora bien, en el supuesto de estar tratando con variables aleatorias asimétricas (es el caso de la Weibull o la exponencial, por citar dos de las más usuales), quizás sería conveniente flexibilizar la ubicación de los límites de precontrol, de manera que no tengan por qué ser simétricos respecto del valor nominal (véase la subsección 7.2.3). Incluso podría pensarse si es más conveniente continuar tomando como referencia el valor nominal (media deseada para la característica de calidad) o bien considerar la mediana de la distribución cuando el proceso se encuentra bajo control (como, de hecho, se ha propuesto en la subsección 6.1.3).

En resumen, la adecuación a otras distribuciones de probabilidad diferentes de la normal requeriría tener en cuenta estos y otros detalles, pero no debería presentar mayores dificultades. Véase la ya referida subsección 6.1.3 y la 7.2.8 a continuación.

7.2.8. Aplicación en el ámbito de la fiabilidad

En la subsección 6.1.3 se ha presentado una primera aproximación a la aplicación del precontrol óptimo a la monitorización de la fiabilidad de sistemas.

La primera mejora que nos planteamos en esta línea es la total flexibilización de los límites de tolerancia y de precontrol, sin necesidad de que sean *simétricos en términos de probabilidad*.

Además, es conveniente realizar una comparación formal de esta técnica con otras herramientas existentes, tales como los mencionados gráficos exponenciales.

Por último, cabe comentar que, en el ejemplo numérico de la subsección 6.1.3, el modelo (6.12) se resolvió mediante la herramienta Solver de Excel. Sabemos por las experiencias llevada a cabo en la subsección 3.5.2 que dicha herramienta no siempre alcanza el resultado óptimo. En este caso, el hecho favorable es que la función de distribución de una variable exponencial no presenta, aparentemente, tantas dificultades de cálculo como la distribución normal. En todo caso, sería deseable estudiar de modo teórico las propiedades del modelo de precontrol óptimo exponencial y comprobar la validez de las soluciones obtenidas, en este caso, con Excel Solver.

7.2.9. Otras estrategias de resolución

Desde un punto de vista matemático, se suele recurrir a enfoques heurísticos y metaheurísticos cuando no es posible (o es excesivamente costoso) abordar un problema de forma *exacta*, utilizando técnicas clásicas. Por otro lado, desde una perspectiva aplicada, de ingeniería, la ventaja de una técnica metaheurística está en su capacidad para generar soluciones *suficientemente buenas* en un periodo de tiempo *aceptable*.

Por tanto, si bien es innecesario aplicar técnicas heurísticas o metaheurísticas a un problema que es resoluble de manera exacta en un tiempo razonable, como es el caso del OPC en su versión actual, la experiencia adquirida con las pruebas realizadas sobre él con la *scatter search* (véase la subsección 6.1.4) nos hacen pensar que puede ser interesante considerarla como técnica de resolución de otros problemas similares pero de mayor complejidad que podamos abordar en el futuro (véanse las subsecciones 7.2.10, 7.2.11 y 7.2.12).

A continuación se exponen las características y ventajas de esta metaheurística que la hacen atractiva desde nuestro punto de vista.

Búsqueda dispersa

La primera descripción del método de *scatter search* se debe a [Glover \(1977\)](#), quien también publica una versión posterior del mismo ([Glover, 1998](#)) más depurada y específica. [Laguna y Marti \(2003\)](#) publican el primer libro monográfico sobre *scatter search*; un resumen muy completo puede encontrarse en [Martí \(2006\)](#).

Como método evolutivo, la búsqueda dispersa basa su funcionamiento en el mantenimiento de un conjunto de soluciones que evoluciona mediante la combinación de éstas. La principal novedad de la búsqueda dispersa respecto a otros esquemas evolutivos (siendo los algoritmos genéticos el paradigma) es que no se fundamenta en la aleatorización sino en decisiones estratégicas y sistemáticas, con las que se consigue mantener el equilibrio entre intensificación y diversificación que debe pretender todo algoritmo metaheurístico en su estrategia de búsqueda de soluciones.

El esquema de búsqueda dispersa se ha aplicado con éxito en un amplio rango problemas combinatorios, pero también en problemas de índole continua. En particular, [Ugray et al. \(2005\)](#) publicaron una implementación de la *scatter search* para problemas no lineales con presencia de variables enteras (MINLP) y en los que todas las funciones involucradas son diferenciables con respecto a las variables continuas. Asimismo, [Egea et al. \(2007\)](#) proponen una *scatter search* para problemas altamente no lineales en los que la función objetivo es una *caja negra* o bien su cálculo (o el del resto de funciones involucradas) resulta muy costoso computacionalmente.

Ambos trabajos son implementaciones de propósito general; son aplicables a problemas de cualquier ámbito que verifiquen unas características mínimas. Otra alternativa que, en ocasiones, resulta más conveniente es adaptar un esquema general (como este de la búsqueda dispersa) al tipo de problema concreto que se quiere resolver, sacando provecho de las particularidades del mismo.

Lo que nosotros nos planteamos, allí donde otros aplican —por ejemplo— algoritmos genéticos, y siempre que no sea viable en la

práctica la búsqueda de una solución óptima exacta, es el diseño de algoritmos específicos basados en el esquema de búsqueda dispersa para la optimización de herramientas de control de la calidad. Las razones que nos conducen a esta elección no son otras que las ya mencionadas: por un lado, los métodos basados en metáforas evolutivas, bien diseñados y calibrados, son capaces de conseguir soluciones *suficientemente buenas* cuando se aplican sobre problemas *complicados*. Por otro lado, la *scatter search*, tanto en la filosofía que la inspira como en su esquema concreto, favorece la inclusión de decisiones y criterios basados en las características concretas del problema a resolver, y no se apoya tanto en la aleatorización como otros métodos metaheurísticos.

7.2.10. Precontrol óptimo multiobjetivo

Como extensión lógica de lo ya dicho en la subsección 7.2.3, otra vía de investigación, muy en la línea de lo que se intenta en otros ámbitos de aplicación de la Investigación Operativa, es abordar el problema del precontrol óptimo desde una perspectiva multiobjetivo (Miettinen, 1999; Ehrgott, 2005).

Estamos pensando, principalmente, en la posibilidad de buscar soluciones que sean simultáneamente tan buenas como sean posibles para varios objetivos más o menos contrapuestos, como podrían ser —por ejemplo— minimizar la tasa de falsa alarma y al mismo tiempo maximizar la potencia para detectar ciertas desviaciones, limitando el tamaño muestral esperado.

De nuevo, como se ha dicho en la subsección 7.2.3, este enfoque requiere evaluar primero qué necesidades prácticas se tienen en relación al precontrol multiobjetivo.

7.2.11. Precontrol óptimo multivariante

Otra extensión lógica de nuestro trabajo es abordar el caso multivariante. Como se ha visto en la subsección 2.3.5, no hay mucha literatura al respecto. Es un campo en el que, primero que nada, se debe plantear si

el enfoque basado en el estadístico T^2 de Hotelling es el más adecuado en la práctica o si, por el contrario, hay que *repensar* cómo generalizar el precontrol al caso multivariante, sin perder de vista la sencillez de la versión univariante. A partir de ahí, ya podría aplicarse un enfoque de optimización similar al desarrollado en la presente memoria.

7.2.12. Extensión a otras herramientas de control estadístico de la calidad

Tal y como se ha destacado en las conclusiones de la presente tesis doctoral, el enfoque *exacto* con que se ha abordado la determinación óptima de los parámetros del precontrol generalizado no es muy frecuente en el campo de la optimización en control de la calidad.

Sería interesante plantearse, por tanto, la optimización de herramientas de control estadístico de procesos desde un punto de vista *exacto*, es decir, sin renunciar de antemano a la obtención de la solución que cumpla de manera óptima con unos requerimientos dados. Ello probablemente implicará, en cada caso, el estudio formal de las propiedades de las funciones involucradas, de modo similar a como se ha hecho en el capítulo 4 de esta memoria.

Si en algún caso se determina que no es viable en la práctica un enfoque exacto, el conocimiento adquirido sobre el problema a resolver a partir de dicho estudio formal sería aprovechado para la elaboración de estrategias heurísticas de resolución ad hoc, como se ha planteado en la subsección 7.2.9.

Anexos

Anexo A

Resultados de la experiencia numérica con Excel y GAMS

N.º ej.*	Parámetros del problema				Mejor solución encontrada				Comportamiento					
	C_p	α^*	β^*	$\delta^{*\dagger}$	k	t	λ	$E[n(0)]$	Mejor sol. (%)‡	Sin sol. (%)§	Error (%)¶	Desv. med. (%)	Desv. 2Q (%)**	Desv. máx. (%)††
1	1,2	0,0027	0,005	1,5462	—	—	—	—	—	100	2	—	—	—
2	1,2	0,0027	0,05	1,5462	—	—	—	—	—	100	1	—	—	—
3	1,2	0,0027	0,1	1,5462	—	—	—	—	—	100	1	—	—	—
4	1,2	0,005	0,005	1,5462	—	—	—	—	—	100	1	—	—	—
5	1,2	0,005	0,05	1,5462	—	—	—	—	—	100	1	—	—	—
6	1,2	0,005	0,1	1,5462	5	6	5,8621	11,1162	58	19	1	5,76	0,00	28,73
7	1,2	0,01	0,005	1,5462	11	5	4,7667	27,8427	69	57	1	1,80	0,00	9,29
8	1,2	0,01	0,05	1,5462	7	4	4,8458	13,0902	51	20	1	13,61	0,00	45,55
9	1,2	0,01	0,1	1,5462	4	6	6,8731	10,2431	57	16	1	9,19	0,00	40,93
10	1,3333	0,0027	0,005	1,9463	7	4	5,0959	11,8100	53	7	1	10,00	0,00	55,03
11	1,3333	0,0027	0,05	1,9463	4	5	5,8981	6,6121	57	7	1	7,20	0,00	47,33
12	1,3333	0,0027	0,1	1,9463	3	5	6,4599	4,9612	79	7	1	5,47	0,00	57,80
13	1,3333	0,005	0,005	1,9463	7	4	5,0959	11,8100	75	13	2	12,05	3,92	67,88
14	1,3333	0,005	0,05	1,9463	4	4	5,6329	6,1923	63	7	1	8,67	0,00	59,88
15	1,3333	0,005	0,1	1,9463	3	5	6,4599	4,9612	77	8	1	3,12	0,00	43,19
16	1,3333	0,01	0,005	1,9463	6	4	5,5957	11,0913	54	7	1	14,58	0,00	69,83
17	1,3333	0,01	0,05	1,9463	3	5	7,3682	5,9089	61	12	1	13,09	0,00	57,61
18	1,3333	0,01	0,1	1,9463	3	4	6,0995	4,6126	61	6	0	7,06	0,00	60,88
19	1,4	0,0027	0,005	2,1463	5	5	5,9710	8,6683	54	7	1	9,49	0,00	57,47
20	1,4	0,0027	0,05	2,1463	3	5	6,5500	4,7553	77	9	1	4,06	0,00	46,08
21	1,4	0,0027	0,1	2,1463	2	7	9,0620	3,9364	63	8	1	3,22	0,00	50,78
22	1,4	0,005	0,005	2,1463	5	4	5,7825	8,2111	55	7	1	12,81	0,00	59,71
23	1,4	0,005	0,05	2,1463	3	4	6,2352	4,4789	69	7	1	5,73	0,00	48,01
24	1,4	0,005	0,1	2,1463	2	6	8,6201	3,7064	71	7	1	3,31	0,00	53,66

Tabla A.1: Relación de problemas de OPC que constituyen la batería de ejemplos objeto de la experiencia numérica de la subsección 3.5.2. Para cada problema, se indica la mejor solución encontrada utilizando los algoritmos seleccionados en dicha sección, así como distintos indicadores de la dificultad de resolución. *Continúa en la página siguiente.*

N.º ej.*	Parámetros del problema				Mejor solución encontrada				Comportamiento					
	C_p	α^*	β^*	$\delta^{*\dagger}$	k	t	λ	$E[n(0)]$	Mejor sol. (%) [‡]	Sin sol. (%) [§]	Error (%) [¶]	Desv. med. (%)	Desv. 2Q (%) ^{**}	Desv. máx. (%) ^{††}
25	1,4	0,01	0,005	2,1463	5	4	5,7826	8,2114	38	7	1	12,96	0,86	50,34
26	1,4	0,01	0,05	2,1463	3	4	6,2352	4,4789	71	13	0	5,47	0,00	49,72
27	1,4	0,01	0,1	2,1463	2	5	8,1312	3,4605	68	7	0	6,01	0,00	56,73
28	1,5	0,0027	0,005	2,4463	4	4	5,9255	5,7056	48	7	1	6,46	1,86	66,87
29	1,5	0,0027	0,05	2,4463	2	5	7,9180	3,1404	86	1	1	1,82	0,00	47,58
30	1,5	0,0027	0,1	2,4463	2	4	6,4010	2,6026	62	7	1	3,47	0,00	39,85
31	1,5	0,005	0,005	2,4463	3	5	7,9018	5,5329	60	8	1	4,60	0,00	31,57
32	1,5	0,005	0,05	2,4463	2	5	7,9179	3,1404	87	7	0	1,96	0,00	47,58
33	1,5	0,005	0,1	2,4463	2	3	5,9627	2,4638	64	7	1	3,86	0,00	41,47
34	1,5	0,01	0,005	2,4463	4	3	5,6996	5,4289	40	16	7	10,51	1,88	86,93
35	1,5	0,01	0,05	2,4463	2	4	7,4774	2,9589	83	6	0	4,77	0,00	50,61
36	1,5	0,01	0,1	2,4463	2	3	5,9628	2,4638	67	7	1	5,51	0,00	52,56

* Número identificador de ejemplo.

† El valor del parámetro δ^* , desviación inaceptable del proceso, viene fijado por el de C_p , de manera que la proporción de piezas fuera de especificaciones bajo dicha desviación sea el 2%; véase la expresión (2.34).

‡ Porcentaje de algoritmos que alcanzaron la mejor solución encontrada, de entre los que proporcionaron alguna solución.

§ Porcentaje de algoritmos que no proporcionaron ninguna solución (por infactibilidad local o por error).

¶ Porcentaje de algoritmos que terminaron con algún tipo de error.

|| Desviación relativa promedio, en las ejecuciones que devolvieron solución. La desviación relativa es calculada según (3.38).

** Mediana de la desviación relativa, en las ejecuciones que devolvieron solución.

†† Máxima desviación relativa observada en las ejecuciones que devolvieron solución.

Tabla A.1: Relación de problemas de OPC que constituyen la batería de ejemplos objeto de la experiencia numérica de la subsección 3.5.2. Para cada problema, se indica la mejor solución encontrada utilizando los algoritmos seleccionados en dicha sección, así como distintos indicadores de la dificultad de resolución.

Algoritmo*	MILP [†]	NLP [‡]	Mejor sol. (%) [§]	Peor sol. (%) [¶]	Sin sol. (%)	Error (%) ^{**}	Fin. tiempo (%) ^{††}	Desviación ^{‡‡}					
								Media (%)	Mín. (%)	Q1 (%)	Mediana (%)	Q3 (%)	Máx. (%)
Solver	—	—	32	0	68	68	—	0,00	0,00	0,00	0,00	0,00	0,00
AlphaECP			38	3	4	0	0	11,29	0,00	0,00	3,92	14,97	86,93
AlphaECP	BDMLP	CONOPT	32	3	0	0	0	13,62	0,00	0,00	9,29	24,72	54,06
AlphaECP	BDMLP	IPOPT	32	6	0	0	0	14,95	0,00	0,00	7,14	26,74	69,83
AlphaECP	BDMLP	LGO	29	6	0	0	0	15,15	0,00	0,00	7,14	26,74	69,83
AlphaECP	BDMLP	MINOS	35	6	0	0	0	14,65	0,00	0,00	7,14	26,74	69,83
AlphaECP	BDMLP	MSNLP	29	6	0	0	0	14,91	0,00	0,00	6,63	26,74	69,83
AlphaECP	BDMLP	PATHNLP	35	6	0	0	0	14,65	0,00	0,00	7,14	26,74	69,83
AlphaECP	BDMLP	SNOPT	35	6	0	0	0	14,37	0,00	0,00	5,81	26,74	69,83
AlphaECP	CBC	CONOPT	6	16	3	0	0	26,22	0,00	6,68	26,95	38,84	69,83
AlphaECP	CBC	IPOPT	16	13	3	0	0	23,02	0,00	4,99	18,24	37,97	69,83
AlphaECP	CBC	LGO	16	13	3	0	0	24,81	0,00	6,10	24,10	34,74	69,83
AlphaECP	CBC	MINOS	16	16	3	0	0	24,07	0,00	4,99	19,63	38,84	69,83
AlphaECP	CBC	MSNLP	26	6	3	0	0	21,56	0,00	1,11	14,14	37,68	68,97
AlphaECP	CBC	PATHNLP	13	16	3	0	0	25,65	0,00	6,30	19,63	47,68	69,83
AlphaECP	CBC	SNOPT	19	16	3	0	0	23,36	0,00	4,90	11,75	45,65	69,83
AlphaECP	CPLEX	CONOPT	48	3	6	0	0	7,90	0,00	0,00	0,00	10,77	52,56
AlphaECP	CPLEX	IPOPT	52	3	6	0	0	7,04	0,00	0,00	0,00	8,07	52,56
AlphaECP	CPLEX	LGO	52	3	3	0	0	8,23	0,00	0,00	0,00	6,60	86,93
AlphaECP	CPLEX	MINOS	48	10	3	0	0	12,33	0,00	0,00	0,00	11,50	86,93
AlphaECP	CPLEX	MSNLP	58	3	3	0	0	8,81	0,00	0,00	0,00	6,29	86,93
AlphaECP	CPLEX	PATHNLP	42	3	6	0	0	8,23	0,00	0,00	1,45	10,77	52,56
AlphaECP	CPLEX	SNOPT	48	3	6	0	0	6,77	0,00	0,00	0,00	8,07	52,56

Tabla A.2: Relación de algoritmos de programación no lineal entera mixta que han sido probados en la experiencia numérica de la subsección 3.5.2. Para cada algoritmo, se muestran diferentes parámetros que miden su comportamiento sobre los 31 problemas de la batería considerados factibles (se excluyen los ejemplos 1 a 5). *Continúa en la página siguiente.*

Algoritmo*	MILP [†]	NLP [‡]	Mejor sol. (%) [§]	Peor sol. (%) [¶]	Sin sol. (%)	Error (%) ^{**}	Fin. tiempo (%) ^{††}	Desviación ^{‡‡}					
								Media (%)	Mín. (%)	Q1 (%)	Mediana (%)	Q3 (%)	Máx. (%)
AlphaECP	GUROBI	CONOPT	45	0	10	6	0	5,03	0,00	0,00	0,72	6,00	27,81
AlphaECP	GUROBI	IPOPT	42	0	6	0	0	5,17	0,00	0,00	1,88	5,92	27,81
AlphaECP	GUROBI	LGO	45	0	6	0	0	4,42	0,00	0,00	0,56	5,81	29,00
AlphaECP	GUROBI	MINOS	48	0	6	0	0	4,79	0,00	0,00	0,00	5,92	27,81
AlphaECP	GUROBI	MSNLP	45	0	6	0	0	3,51	0,00	0,00	0,57	4,93	23,54
AlphaECP	GUROBI	PATHNLP	42	0	6	0	0	5,83	0,00	0,00	1,88	6,26	27,81
AlphaECP	GUROBI	SNOPT	45	0	6	0	0	5,90	0,00	0,00	1,45	6,62	27,81
AlphaECP	MOSEK	CONOPT	26	0	3	0	0	12,70	0,00	0,36	7,69	17,04	62,99
AlphaECP	MOSEK	IPOPT	29	0	3	0	0	12,56	0,00	0,00	8,58	14,19	62,99
AlphaECP	MOSEK	LGO	39	0	3	0	0	10,72	0,00	0,00	4,23	13,78	62,99
AlphaECP	MOSEK	MINOS	26	0	3	0	0	13,20	0,00	0,36	9,01	17,04	62,99
AlphaECP	MOSEK	MSNLP	45	0	3	0	0	8,10	0,00	0,00	2,24	11,64	45,63
AlphaECP	MOSEK	PATHNLP	23	0	3	0	0	13,39	0,00	1,84	9,01	17,62	62,99
AlphaECP	MOSEK	SNOPT	29	0	3	0	0	12,43	0,00	0,00	7,03	17,04	62,99
AlphaECP	SULUM	CONOPT	55	0	3	0	0	4,18	0,00	0,00	0,00	3,19	45,03
AlphaECP	SULUM	IPOPT	55	0	3	0	0	3,84	0,00	0,00	0,00	3,19	45,03
AlphaECP	SULUM	LGO	52	0	3	0	0	3,59	0,00	0,00	0,00	3,77	45,03
AlphaECP	SULUM	MINOS	52	0	3	0	0	3,58	0,00	0,00	0,00	3,77	45,03
AlphaECP	SULUM	MSNLP	52	0	3	0	0	3,15	0,00	0,00	0,00	3,19	45,03
AlphaECP	SULUM	PATHNLP	55	0	3	0	0	2,95	0,00	0,00	0,00	2,56	45,03
AlphaECP	SULUM	SNOPT	55	0	3	0	0	3,39	0,00	0,00	0,00	3,19	45,03
AlphaECP	XA	CONOPT	42	0	6	0	0	4,71	0,00	0,00	0,86	7,14	29,00
AlphaECP	XA	IPOPT	39	0	6	0	0	5,32	0,00	0,00	3,33	9,54	29,00
AlphaECP	XA	LGO	45	0	6	0	0	2,99	0,00	0,00	0,86	5,81	12,10

Tabla A.2: Relación de algoritmos de programación no lineal entera mixta que han sido probados en la experiencia numérica de la subsección 3.5.2. Para cada algoritmo, se muestran diferentes parámetros que miden su comportamiento sobre los 31 problemas de la batería considerados factibles (se excluyen los ejemplos 1 a 5). *Continúa en la página siguiente.*

Algoritmo*	MILP [†]	NLP [‡]	Mejor sol. (%) [§]	Peor sol. (%) [¶]	Sin sol. (%)	Error (%) ^{**}	Fin. tiempo (%) ^{††}	Desviación ^{‡‡}					
								Media (%)	Mín. (%)	Q1 (%)	Mediana (%)	Q3 (%)	Máx. (%)
AlphaECP	XA	MINOS	42	0	6	0	0	4,71	0,00	0,00	0,86	7,14	29,00
AlphaECP	XA	MSNLP	45	0	6	0	0	4,25	0,00	0,00	0,86	6,59	29,00
AlphaECP	XA	PATHNLP	42	0	6	0	0	4,71	0,00	0,00	0,86	7,14	29,00
AlphaECP	XA	SNOPT	42	0	6	0	0	4,70	0,00	0,00	0,86	7,14	29,00
AlphaECP	XPRESS	CONOPT	32	6	3	0	0	19,17	0,00	0,00	9,54	36,65	57,80
AlphaECP	XPRESS	IPOPT	26	3	3	0	0	19,31	0,00	0,36	18,66	34,41	55,03
AlphaECP	XPRESS	LGO	42	0	3	0	0	15,06	0,00	0,00	4,13	29,29	55,03
AlphaECP	XPRESS	MINOS	32	3	3	0	0	18,07	0,00	0,00	9,54	34,41	55,03
AlphaECP	XPRESS	MSNLP	42	0	3	0	0	14,88	0,00	0,00	6,61	27,29	61,14
AlphaECP	XPRESS	PATHNLP	32	6	3	0	0	19,17	0,00	0,00	9,54	36,65	57,80
AlphaECP	XPRESS	SNOPT	32	3	3	0	0	18,08	0,00	0,00	9,54	34,41	55,03
Bon. B-BB	—	—	84	0	0	0	0	0,31	0,00	0,00	0,00	0,00	5,33
Bon. B-ECP	—	—	23	3	45	45	0	13,69	0,00	0,00	11,14	28,53	57,61
Bon. B-Hyb	—	—	10	35	6	0	0	35,27	0,00	27,36	40,44	49,72	56,73
Bon. B-iFP	—	—	0	42	6	0	0	41,75	6,00	37,49	43,52	50,61	60,88
Bon. B-OA	—	—	32	0	0	0	0	12,26	0,00	0,00	5,81	21,25	50,50
Bon. B-QG	—	—	0	0	3	0	0	16,59	9,75	13,78	17,29	18,74	24,20
DICOPT			80	0	18	0	0	0,00	0,00	0,00	0,00	0,00	0,86
DICOPT	BDMLP	CONOPT	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	BDMLP	IPOPT	84	0	13	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	BDMLP	LGO	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	BDMLP	MINOS	90	0	6	3	3	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	BDMLP	MSNLP	3	0	97	0	0	0,00	0,00	0,00	0,00	0,00	0,00

Tabla A.2: Relación de algoritmos de programación no lineal entera mixta que han sido probados en la experiencia numérica de la subsección 3.5.2. Para cada algoritmo, se muestran diferentes parámetros que miden su comportamiento sobre los 31 problemas de la batería considerados factibles (se excluyen los ejemplos 1 a 5). *Continúa en la página siguiente.*

Algoritmo*	MILP [†]	NLP [‡]	Mejor sol. (%) [§]	Peor sol. (%) [¶]	Sin sol. (%)	Error (%) ^{**}	Fin. tiempo (%) ^{††}	Desviación ^{‡‡}					
								Media (%)	Mín. (%)	Q1 (%)	Mediana (%)	Q3 (%)	Máx. (%)
DICOPT	BDMLP	PATHNLP	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	BDMLP	SNOPT	100	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CBC	CONOPT	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CBC	IPOPT	84	0	13	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CBC	LGO	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CBC	MINOS	90	0	6	3	3	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CBC	MSNLP	3	0	97	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CBC	PATHNLP	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CBC	SNOPT	100	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CPLEX	CONOPT	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CPLEX	IPOPT	84	0	13	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CPLEX	LGO	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CPLEX	MINOS	90	0	6	3	3	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CPLEX	MSNLP	3	0	97	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CPLEX	PATHNLP	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	CPLEX	SNOPT	100	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	GUROBI	CONOPT	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	GUROBI	IPOPT	84	0	13	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	GUROBI	LGO	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	GUROBI	MINOS	90	0	6	3	3	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	GUROBI	MSNLP	3	0	97	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	GUROBI	PATHNLP	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	GUROBI	SNOPT	100	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	MOSEK	CONOPT	97	0	3	0	0	0,00	0,00	0,00	0,00	0,00	0,00

Tabla A.2: Relación de algoritmos de programación no lineal entera mixta que han sido probados en la experiencia numérica de la subsección 3.5.2. Para cada algoritmo, se muestran diferentes parámetros que miden su comportamiento sobre los 31 problemas de la batería considerados factibles (se excluyen los ejemplos 1 a 5). *Continúa en la página siguiente.*

Algoritmo*	MILP [†]	NLP [‡]	Mejor sol. (%) [§]	Peor sol. (%) [¶]	Sin sol. (%)	Error (%) ^{**}	Fin. tiempo (%) ^{††}	Desviación ^{‡‡}					
								Media (%)	Mín. (%)	Q1 (%)	Mediana (%)	Q3 (%)	Máx. (%)
DICOPT	MOSEK	IPOPT	84	0	13	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	MOSEK	LGO	90	0	3	0	0	0,03	0,00	0,00	0,00	0,00	0,81
DICOPT	MOSEK	MINOS	84	0	13	3	3	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	MOSEK	MSNLP	3	0	97	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	MOSEK	PATHNLP	94	0	3	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	MOSEK	SNOPT	94	0	6	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	SULUM	CONOPT	90	0	3	0	0	0,03	0,00	0,00	0,00	0,00	0,86
DICOPT	SULUM	IPOPT	77	0	19	0	0	0,03	0,00	0,00	0,00	0,00	0,86
DICOPT	SULUM	LGO	90	0	3	0	0	0,03	0,00	0,00	0,00	0,00	0,86
DICOPT	SULUM	MINOS	84	0	13	3	3	0,03	0,00	0,00	0,00	0,00	0,86
DICOPT	SULUM	MSNLP	3	0	97	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	SULUM	PATHNLP	90	0	6	0	0	0,03	0,00	0,00	0,00	0,00	0,86
DICOPT	SULUM	SNOPT	87	0	10	0	0	0,03	0,00	0,00	0,00	0,00	0,86
DICOPT	XA	CONOPT	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	XA	IPOPT	84	0	13	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	XA	LGO	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	XA	MINOS	90	0	6	3	3	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	XA	MSNLP	3	0	97	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	XA	PATHNLP	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	XA	SNOPT	100	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	XPRESS	CONOPT	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	XPRESS	IPOPT	84	0	13	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	XPRESS	LGO	94	0	3	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	XPRESS	MINOS	90	0	6	3	3	0,00	0,00	0,00	0,00	0,00	0,00

Tabla A.2: Relación de algoritmos de programación no lineal entera mixta que han sido probados en la experiencia numérica de la subsección 3.5.2. Para cada algoritmo, se muestran diferentes parámetros que miden su comportamiento sobre los 31 problemas de la batería considerados factibles (se excluyen los ejemplos 1 a 5). *Continúa en la página siguiente.*

Algoritmo*	MILP [†]	NLP [‡]	Mejor sol. (%) [§]	Peor sol. (%) [¶]	Sin sol. (%)	Error (%) ^{**}	Fin. tiempo (%) ^{††}	Desviación ^{‡‡}					
								Media (%)	Mín. (%)	Q1 (%)	Mediana (%)	Q3 (%)	Máx. (%)
DICOPT	XPRESS	MSNLP	3	0	97	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	XPRESS	PATHNLP	97	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
DICOPT	XPRESS	SNOPT	100	0	0	0	0	0,00	0,00	0,00	0,00	0,00	0,00
KNITRO	—	—	74	0	0	0	0	0,73	0,00	0,00	0,00	0,00	6,84
LINDOG.	—	—	61	0	0	0	100	1,01	0,00	0,00	0,00	1,49	6,20
OQNLP			17	2	3	0	9	19,01	0,00	6,38	22,08	27,65	59,71
OQNLP	—	CONOPT	19	0	3	0	0	15,76	0,00	4,10	17,24	25,53	34,75
OQNLP	—	IPOPT	19	0	3	0	3	16,29	0,00	4,10	17,24	26,30	41,46
OQNLP	—	LGO	19	0	3	0	0	16,62	0,00	6,83	20,00	26,30	34,75
OQNLP	—	MINOS	3	13	3	0	58	31,81	0,00	23,72	32,03	40,12	59,71
OQNLP	—	MSNLP	16	3	3	0	0	21,10	0,00	5,57	24,24	30,07	54,66
OQNLP	—	PATHNLP	19	0	3	0	0	15,76	0,00	4,10	17,24	25,53	34,75
OQNLP	—	SNOPT	19	0	3	0	0	15,76	0,00	4,10	17,24	25,53	34,75
SBB			64	0	6	0	7	0,88	0,00	0,00	0,00	0,86	23,85
SBB	—	CONOPT	68	0	0	0	0	0,77	0,00	0,00	0,00	0,71	5,33
SBB	—	IPOPT	68	0	0	0	0	0,77	0,00	0,00	0,00	0,71	5,33
SBB	—	LGO	68	0	0	0	0	0,77	0,00	0,00	0,00	0,71	5,33
SBB	—	MINOS	39	0	42	0	48	1,99	0,00	0,00	0,00	0,78	23,85
SBB	—	MSNLP	68	0	0	0	0	0,77	0,00	0,00	0,00	0,71	5,33
SBB	—	PATHNLP	68	0	0	0	0	0,79	0,00	0,00	0,00	0,83	5,33
SBB	—	SNOPT	68	0	0	0	0	0,77	0,00	0,00	0,00	0,71	5,33

Tabla A.2: Relación de algoritmos de programación no lineal entera mixta que han sido probados en la experiencia numérica de la subsección 3.5.2. Para cada algoritmo, se muestran diferentes parámetros que miden su comportamiento sobre los 31 problemas de la batería considerados factibles (se excluyen los ejemplos 1 a 5). *Continúa en la página siguiente.*

Algoritmo*	MILP [†]	NLP [‡]	Mejor sol. (%) [§]	Peor sol. (%) [¶]	Sin sol. (%)	Error (%) ^{**}	Fin. tiempo (%) ^{††}	Desviación ^{‡‡}				
								Media (%)	Mín. (%)	Q1 (%)	Mediana (%)	Q3 (%)

* Véase más información sobre los algoritmos en la tabla 3.1.

[†] Técnica auxiliar de programación lineal entera que emplea el algoritmo, cuando admite más de una. Véase más información en la tabla 3.2.

[‡] Técnica auxiliar de programación no lineal que emplea el algoritmo, cuando admite más de una. Véase más información en la tabla 3.3.

[§] Porcentaje de ejecuciones en que el algoritmo alcanzó la mejor solución conocida, respecto del total de ejecuciones.

[¶] Porcentaje de ejecuciones en que el algoritmo proporcionó la peor solución devuelta por algún algoritmo, respecto del total de ejecuciones.

^{||} Porcentaje de ejecuciones en que el algoritmo no proporcionó ninguna solución (bien por infactibilidad local, o bien por terminar con algún tipo de error), respecto del total de ejecuciones.

^{**} Porcentaje de ejecuciones en que el algoritmo terminó con algún tipo de error, respecto del total de ejecuciones.

^{††} Porcentaje de ejecuciones en que el algoritmo finalizó por consumir el tiempo disponible, respecto del total de ejecuciones.

^{‡‡} Estadísticos descriptivos de la desviación para cada algoritmo (calculada según (3.38)), respecto del total de ejecuciones que devolvieron solución. Media = Valor promedio. Mín. = Valor mínimo observado. Q1 = Primer cuartil. Mediana = Valor de la mediana. Q3 = Tercer cuartil. Máx. = Valor máximo observado.

Tabla A.2: Relación de algoritmos de programación no lineal entera mixta que han sido probados en la experiencia numérica de la subsección 3.5.2. Para cada algoritmo, se muestran diferentes parámetros que miden su comportamiento sobre los 31 problemas de la batería considerados factibles (se excluyen los ejemplos 1 a 5).

Fuente de variabilidad	Suma de cuadrados	g.l.	Cuadrado medio	Cociente F	p -valor
Efectos principales					
MILP	6,6507	7	0,9501	68,38	0,0000
NLP	0,15231	6	0,0253851	1,83	0,0904
C_p	1,38011	2	0,690054	49,66	0,0000
α^*	1,41516	2	0,70758	50,93	0,0000
β^*	5,07075	2	2,53537	182,47	0,0000
Interacciones					
MILP · NLP	0,170595	42	0,00406179	0,29	1,0000
MILP · C_p	7,33321	14	0,5238	37,7	0,0000
MILP · α^*	1,85698	14	0,132641	9,55	0,0000
MILP · β^*	2,72573	14	0,194695	14,01	0,0000
NLP · C_p	0,0336005	12	0,00280004	0,2	0,9984
NLP · α^*	0,0594662	12	0,00495551	0,36	0,9776
NLP · β^*	0,0637499	12	0,00531249	0,38	0,9701
C_p · α^*	0,325968	4	0,081492	5,87	0,0001
C_p · β^*	0,884743	4	0,221186	15,92	0,0000
α^* · β^*	0,24858	4	0,0621449	4,47	0,0014
Residuos	18,8965	1360	0,0138945		
Total (corregido)	47,2682	1511			

Tabla A.3: ANOVA correspondiente a la calibración del algoritmo GAMS/AlphaECP, para la experiencia numérica de la subsección 3.5.2.

Nivel	Núm. de obs.	Desviación media (%)	Nivel	Núm. de obs.	Desviación media (%)
MILP			NLP		
BDMLP	189	13,48	CONOPT	216	12,2883
CBC	189	23,0282	IPOPT	216	11,5249
CPLEX	189	10,9433	LGO	216	10,1377
GUROBI	189	5,64731	MINOS	216	11,5378
MOSEK	189	13,1905	MSNLP	216	9,22945
SULUM	189	2,24883	PATHNLP	216	12,0155
XA	189	3,55951	SNOPT	216	11,213
XPRESS	189	16,9842			

Tabla A.4: Tabla de desviaciones medias para cada nivel de los métodos auxiliares de MILP y de NLP correspondiente al algoritmo GAMS/AlphaECP, para la experiencia numérica de la subsección 3.5.2.

Nivel	N. de obs.	Desv. media (%)	Nivel	N. de obs.	Desv. media (%)	Nivel	N. de obs.	Desv. media (%)
C_p			α^*			β^*		
1,3333	504	15,1333	0,0027	504	8,35276	0,005	504	19,2459
1,4	504	10,4413	0,005	504	9,65721	0,05	504	8,06349
1,5	504	7,83109	0,01	504	15,3957	0,1	504	6,09628

Tabla A.5: Tabla de desviaciones medias para cada nivel de los parámetros C_p , α^* y β^* correspondiente al algoritmo GAMS/AlphaECP, para la experiencia numérica de la subsección 3.5.2.

Combinación	N. de obs.	Desviación media (%)	Combinación	N. de obs.	Desviación media (%)
BDMLP,CONOPT	27	12,3426	MOSEK,CONOPT	27	14,114
BDMLP,IPOPT	27	13,8676	MOSEK,IPOPT	27	13,9515
BDMLP,LGO	27	14,1007	MOSEK,LGO	27	11,9102
BDMLP,MINOS	27	13,5173	MOSEK,MINOS	27	14,6655
BDMLP,MSNLP	27	13,8164	MOSEK,MSNLP	27	9,00377
BDMLP,PATHNLP	27	13,5173	MOSEK,PATHNLP	27	14,877
BDMLP,SNOPT	27	13,1983	MOSEK,SNOPT	27	13,8115
CBC,CONOPT	27	25,2601	SULUM,CONOPT	27	2,97622
CBC,IPOPT	27	21,7024	SULUM,IPOPT	27	2,59552
CBC,LGO	27	23,9893	SULUM,LGO	27	2,32095
CBC,MINOS	27	22,8633	SULUM,MINOS	27	2,30993
CBC,MSNLP	27	20,4274	SULUM,MSNLP	27	1,8357
CBC,PATHNLP	27	24,8809	SULUM,PATHNLP	27	1,60927
CBC,SNOPT	27	22,0741	SULUM,SNOPT	27	2,09423
CPLEX,CONOPT	27	12,2351	XA,CONOPT	27	3,71517
CPLEX,IPOPT	27	11,3144	XA,IPOPT	27	4,38053
CPLEX,LGO	27	8,06478	XA,LGO	27	2,94553
CPLEX,MINOS	27	12,6254	XA,MINOS	27	3,71517
CPLEX,MSNLP	27	8,74998	XA,MSNLP	27	2,73046
CPLEX,PATHNLP	27	12,5911	XA,PATHNLP	27	3,71517
CPLEX,SNOPT	27	11,0225	XA,SNOPT	27	3,7145
GUROBI,CONOPT	27	8,98984	XPRESS,CONOPT	27	18,6734
GUROBI,IPOPT	27	5,54785	XPRESS,IPOPT	27	18,8391
GUROBI,LGO	27	3,66965	XPRESS,LGO	27	14,1007
GUROBI,MINOS	27	5,14849	XPRESS,MINOS	27	17,4571
GUROBI,MSNLP	27	3,5832	XPRESS,MSNLP	27	13,6886
GUROBI,PATHNLP	27	6,25994	XPRESS,PATHNLP	27	18,6734
GUROBI,SNOPT	27	6,33216	XPRESS,SNOPT	27	17,4567

Tabla A.6: Tabla de desviaciones medias para cada combinación de niveles de los métodos auxiliares de MILP y NLP correspondiente al algoritmo GAMS/AlphaECP, para la experiencia numérica de la subsección 3.5.2.

Fuente de variabilidad	Suma de cuadrados	g.l.	Cuadrado medio	Cociente F	p -valor
Efectos principales					
MILP	0,399362	7	0,0570517	13,04	0,0000
NLP	107,992	6	17,9987	4114,87	0,0000
U. exper.	6,65268	26	0,255872	58,5	0,0000
Interacciones					
MILP · NLP	0,178004	42	0,00423819	0,97	0,5288
MILP · u. exper.	5,17994	182	0,0284612	6,51	0,0000
NLP · u. exper.	21,9931	156	0,140982	32,23	0,0000
Residuos	4,77647	1092	0,00437406		
Total (corregido)	147,172	1511			

Tabla A.7: ANOVA correspondiente a la calibración del algoritmo GAMS/DICOPT, para la experiencia numérica de la subsección 3.5.2.

Nivel	Núm. de obs.	Desviación media (%)	Nivel	Núm. de obs.	Desviación media (%)
MILP			NLP		
BDMLP	189	12,2702	CONOPT	216	0,875505
CBC	189	12,2702	IPOPT	216	0,875505
CPLEX	189	12,2702	LGO	216	1,28918
GUROBI	189	12,2702	MINOS	216	8,6861
MOSEK	189	13,9947	MSNLP	216	78,2994
SULUM	189	17,1957	PATHNLP	216	0,875506
XA	189	12,2702	SNOPT	216	1,20886
XPRESS	189	12,7274			

Tabla A.8: Tabla de desviaciones medias para cada nivel de los métodos auxiliares de MILP y de NLP correspondiente al algoritmo GAMS/DICOPT, para la experiencia numérica de la subsección 3.5.2.

U. exp.	N. de obs.	Desv. media (%)	U. exp.	N. de obs.	Desv. media (%)	U. exp.	N. de obs.	Desv. media (%)
10	56	11,7929	19	56	12,3143	28	56	14,3301
11	56	10,1426	20	56	9,87341	29	56	0
12	56	12,3866	21	56	10,882	30	56	8,53957
13	56	23,6379	22	56	12,795	31	56	7,61066
14	56	12,8324	23	56	11,5742	32	56	10,196
15	56	9,25414	24	56	11,4983	33	56	8,88686
16	56	16,8352	25	56	10,8782	34	56	37,2552
17	56	23,148	26	56	21,3067	35	56	10,845
18	56	13,0459	27	56	12,1571	36	56	11,2631

Tabla A.9: Tabla de desviaciones medias para cada unidad experimental correspondiente al algoritmo GAMS/DICOPT, para la experiencia numérica de la subsección [3.5.2.](#)

Combinación	N. obs.	Desviación media (%)	Combinación	N. obs.	Desviación media (%)
BDMLP,CONOPT	27	$6,15325 \cdot 10^{-5}$	MOSEK,CONOPT	27	3,77112
BDMLP,IPOPT	27	$6,15223 \cdot 10^{-5}$	MOSEK,IPOPT	27	$6,15223 \cdot 10^{-5}$
BDMLP,LGO	27	$6,15325 \cdot 10^{-5}$	MOSEK,LGO	27	3,87978
BDMLP,MINOS	27	7,59142	MOSEK,MINOS	27	9,34532
BDMLP,MSNLP	27	78,2994	MOSEK,MSNLP	27	78,2994
BDMLP,PATHNLP	27	$6,72696 \cdot 10^{-5}$	MOSEK,PATHNLP	27	$6,72696 \cdot 10^{-5}$
BDMLP,SNOPT	27	0	MOSEK,SNOPT	27	2,66731
CBC,CONOPT	27	$6,15325 \cdot 10^{-5}$	SULUM,CONOPT	27	3,23255
CBC,IPOPT	27	$6,15223 \cdot 10^{-5}$	SULUM,IPOPT	27	7,00361
CBC,LGO	27	$6,15325 \cdot 10^{-5}$	SULUM,LGO	27	3,23255
CBC,MINOS	27	7,59142	SULUM,MINOS	27	14,595
CBC,MSNLP	27	78,2994	SULUM,MSNLP	27	78,2994
CBC,PATHNLP	27	$6,72696 \cdot 10^{-5}$	SULUM,PATHNLP	27	7,00358
CBC,SNOPT	27	0	SULUM,SNOPT	27	7,00354
CPLEX,CONOPT	27	$6,15325 \cdot 10^{-5}$	XA,CONOPT	27	$6,15325 \cdot 10^{-5}$
CPLEX,IPOPT	27	$6,15223 \cdot 10^{-5}$	XA,IPOPT	27	$6,15223 \cdot 10^{-5}$
CPLEX,LGO	27	$6,15325 \cdot 10^{-5}$	XA,LGO	27	$6,15325 \cdot 10^{-5}$
CPLEX,MINOS	27	7,59142	XA,MINOS	27	7,59142
CPLEX,MSNLP	27	78,2994	XA,MSNLP	27	78,2994
CPLEX,PATHNLP	27	$6,72696 \cdot 10^{-5}$	XA,PATHNLP	27	$6,72696 \cdot 10^{-5}$
CPLEX,SNOPT	27	0	XA,SNOPT	27	0
GUROBI,CONOPT	27	$6,15325 \cdot 10^{-5}$	XPRESS,CONOPT	27	$6,15325 \cdot 10^{-5}$
GUROBI,IPOPT	27	$6,15223 \cdot 10^{-5}$	XPRESS,IPOPT	27	$6,15223 \cdot 10^{-5}$
GUROBI,LGO	27	$6,15325 \cdot 10^{-5}$	XPRESS,LGO	27	3,20077
GUROBI,MINOS	27	7,59142	XPRESS,MINOS	27	7,59142
GUROBI,MSNLP	27	78,2994	XPRESS,MSNLP	27	78,2994
GUROBI,PATHNLP	27	$6,72696 \cdot 10^{-5}$	XPRESS,PATHNLP	27	$6,72696 \cdot 10^{-5}$
GUROBI,SNOPT	27	0	XPRESS,SNOPT	27	0

Tabla A.10: Tabla de desviaciones medias para cada combinación de niveles de los métodos auxiliares de MILP y NLP correspondiente al algoritmo GAMS/DICOPT, para la experiencia numérica de la subsección 3.5.2.

Fuente de variabilidad	Suma de cuadrados	g.l.	Cuadrado medio	Cociente F	p -valor
Efectos principales					
NLP	0,717014	6	0,119502	15,5	0,0000
C_p	0,265227	2	0,132613	17,2	0,0000
α^*	0,0494286	2	0,0247143	3,21	0,0438
β^*	0,182622	2	0,091311	11,85	0,0000
Interacciones					
NLP $\cdot C_p$	0,162145	12	0,0135121	1,75	0,0630
NLP $\cdot \alpha^*$	0,0107576	12	0,000896471	0,12	0,9999
NLP $\cdot \beta^*$	0,0384566	12	0,00320472	0,42	0,9551
$C_p \cdot \alpha^*$	0,0251941	4	0,00629852	0,82	0,5165
$C_p \cdot \beta^*$	1,03499	4	0,258748	33,57	0,0000
$\alpha^* \cdot \beta^*$	0,0874451	4	0,0218613	2,84	0,0271
Residuos	0,986611	128	0,0077079		
Total (corregido)	3,55989	188			

Tabla A.11: ANOVA correspondiente a la calibración del algoritmo GAMS/OQNLP, para la experiencia numérica de la subsección 3.5.2.

Nivel	Núm. de obs.	Desviación media (%)
CONOPT	27	16,3308
IPOPT	27	16,3308
LGO	27	17,2911
MINOS	27	34,1726
MSNLP	27	21,9338
PATHNLP	27	16,3308
SNOPT	27	16,3306

Tabla A.12: Tabla de desviaciones medias para cada nivel del método auxiliar de NLP correspondiente al algoritmo GAMS/OQNLP, para la experiencia numérica de la subsección 3.5.2.

Nivel	N. de obs.	Desv. media (%)	Nivel	N. de obs.	Desv. media (%)	Nivel	N. de obs.	Desv. media (%)
C_p			α^*			β^*		
1,3333	63	20,4381	0,0027	63	20,7068	0,005	63	22,1285
1,4	63	24,0631	0,005	63	17,5478	0,05	63	21,9
1,5	63	14,9504	0,01	63	21,1971	0,1	63	15,4232

Tabla A.13: Tabla de desviaciones medias para cada nivel de los parámetros C_p , α^* y β^* correspondiente al algoritmo GAMS/OQNLP, para la experiencia numérica de la subsección 3.5.2.

Fuente de variabilidad	Suma de cuadrados	g.l.	Cuadrado medio	Cociente F	p -valor
Efectos principales					
NLP	0,0665351	6	0,0110892	0,78	0,5906
C_p	0,927067	2	0,463534	32,42	0,0000
α^*	0,802445	2	0,401223	28,06	0,0000
β^*	0,240059	2	0,12003	8,39	0,0004
Interacciones					
NLP · C_p	0,13307	12	0,0110892	0,78	0,6746
NLP · α^*	0,121774	12	0,0101478	0,71	0,7398
NLP · β^*	0,0874199	12	0,00728499	0,51	0,9055
C_p · α^*	1,60489	4	0,401223	28,06	0,0000
C_p · β^*	0,480119	4	0,12003	8,39	0,0000
α^* · β^*	0,437979	4	0,109495	7,66	0,0000
Residuos	1,83026	128	0,0142989		
Total (corregido)	6,73162	188			

Tabla A.14: ANOVA correspondiente a la calibración del algoritmo GAMS/SBB, para la experiencia numérica de la subsección 3.5.2.

Nivel	Núm. de obs.	Desviación media (%)
CONOPT	27	3,08165
IPOPT	27	3,34314
LGO	27	3,284
MINOS	27	4,41217
MSNLP	27	6,43733
PATHNLP	27	5,56421
SNOPT	27	8,54384

Tabla A.15: Tabla de desviaciones medias para cada nivel del método auxiliar de NLP correspondiente al algoritmo GAMS/SBB, para la experiencia numérica de la subsección 3.5.2.

Nivel	N. de obs.	Desv. media (%)	Nivel	N. de obs.	Desv. media (%)	Nivel	N. de obs.	Desv. media (%)
C_p			α^*			β^*		
1,3333	63	0	0,0027	63	0,0897938	0,005	63	0,642094
1,4	63	0	0,005	63	0,604741	0,05	63	4,84499
1,5	63	14,857	0,01	63	14,1625	0,1	63	9,36991

Tabla A.16: Tabla de desviaciones medias para cada nivel de los parámetros C_p , α^* y β^* correspondiente al algoritmo GAMS/SBB, para la experiencia numérica de la subsección 3.5.2.

Fuente de variabilidad	Suma de cuadrados	g.l.	Cuadrado medio	Cociente F	p -valor
Efectos principales					
Algoritmo	12,1099	12	1,00916	38,38	0,0000
C_p	0,142198	2	0,071099	2,7	0,0689
α^*	0,0354843	2	0,0177421	0,67	0,5102
β^*	0,0612996	2	0,0306498	1,17	0,3134
Interacciones					
Alg. $\cdot C_p$	0,476002	24	0,0198334	0,75	0,7914
Alg. $\cdot \alpha^*$	1,35858	24	0,0566077	2,15	0,0019
Alg. $\cdot \beta^*$	1,74578	24	0,0727409	2,77	0,0000
$C_p \cdot \alpha^*$	0,0530314	4	0,0132578	0,5	0,7327
$C_p \cdot \beta^*$	0,038588	4	0,00964701	0,37	0,8321
$\alpha^* \cdot \beta^*$	0,0739193	4	0,0184798	0,7	0,5907
Residuos	6,52131	248	0,0262956		
Total (corregido)	22,6161	350			

Tabla A.17: ANOVA correspondiente a la comparación de los algoritmos considerados en la experiencia numérica de la subsección 3.5.2.

Fuente de variabilidad	Suma de cuadrados	g.l.	Cuadrado medio	Cociente F	p -valor
Efectos principales					
MILP	1,53503	7	0,219291	180,03	0,0000
NLP	0,000927813	6	0,000154636	0,13	0,9928
U. exper.	0,826622	3	0,275541	226,21	0,0000
Interacciones					
MILP · NLP	0,0448843	42	0,00106867	0,88	0,6807
MILP · u. exper.	4,1118	21	0,1958	160,75	0,0000
NLP · u. exper.	0,00722445	18	0,000401358	0,33	0,9955
Residuos	0,153475	126	0,00121806		
Total (corregido)	6,67997	223			

Tabla A.18: ANOVA correspondiente a la calibración del algoritmo GAMS/AlphaECP en el caso $C_p = 1,2$, para la experiencia numérica de la subsección 3.5.2.

Nivel	N. obs.	Desv. media (%)	Nivel	N. obs.	Desv. media (%)	Nivel	N. obs.	Desv. media (%)
MILP			NLP			U. exper.		
BDMLP	28	22,264	CONOPT	32	18,2397	6	56	14,4122
CBC	28	28,7743	IPOPT	32	18,2397	7	56	13,3503
CPLEX	28	10,708	LGO	32	17,9987	8	56	28,5575
GUROBI	28	15,4664	MINOS	32	18,2397	9	56	16,6282
MOSEK	28	3,48273	MSNLP	32	18,6699			
SULUM	28	14,7508	PATHNLP	32	18,0235			
XA	28	29,0399	SNOPT	32	18,2481			
XPRESS	28	21,4102						

Tabla A.19: Tabla de desviaciones medias para cada nivel de los métodos auxiliares de MILP y de NLP y cada unidad experimental correspondiente al algoritmo GAMS/AlphaECP en el caso $C_p = 1,2$, para la experiencia numérica de la subsección 3.5.2.

Combinación	N. de obs.	Desviación media (%)	Combinación	N. de obs.	Desviación media (%)
BDMLP,CONOPT	4	22,264	MOSEK,CONOPT	4	3,48273
BDMLP,IPOPT	4	22,264	MOSEK,IPOPT	4	3,48273
BDMLP,LGO	4	22,264	MOSEK,LGO	4	3,48273
BDMLP,MINOS	4	22,264	MOSEK,MINOS	4	3,48273
BDMLP,MSNLP	4	22,2643	MOSEK,MSNLP	4	3,48273
BDMLP,PATHNLP	4	22,264	MOSEK,PATHNLP	4	3,48273
BDMLP,SNOPT	4	22,2638	MOSEK,SNOPT	4	3,48273
CBC,CONOPT	4	29,647	SULUM,CONOPT	4	14,7508
CBC,IPOPT	4	29,647	SULUM,IPOPT	4	14,7508
CBC,LGO	4	27,6512	SULUM,LGO	4	14,7508
CBC,MINOS	4	29,647	SULUM,MINOS	4	14,7508
CBC,MSNLP	4	27,2636	SULUM,MSNLP	4	14,7509
CBC,PATHNLP	4	27,9175	SULUM,PATHNLP	4	14,7508
CBC,SNOPT	4	29,6469	SULUM,SNOPT	4	14,7506
CPLEX,CONOPT	4	10,7422	XA,CONOPT	4	29,5996
CPLEX,IPOPT	4	10,7422	XA,IPOPT	4	29,5996
CPLEX,LGO	4	10,7422	XA,LGO	4	22,3494
CPLEX,MINOS	4	10,7422	XA,MINOS	4	29,5996
CPLEX,MSNLP	4	10,503	XA,MSNLP	4	32,9317
CPLEX,PATHNLP	4	10,7422	XA,PATHNLP	4	29,5996
CPLEX,SNOPT	4	10,7422	XA,SNOPT	4	29,5996
GUROBI,CONOPT	4	14,2547	XPRESS,CONOPT	4	21,1764
GUROBI,IPOPT	4	14,2547	XPRESS,IPOPT	4	21,1764
GUROBI,LGO	4	21,5049	XPRESS,LGO	4	21,2446
GUROBI,MINOS	4	14,2547	XPRESS,MINOS	4	21,1764
GUROBI,MSNLP	4	15,4866	XPRESS,MSNLP	4	22,6767
GUROBI,PATHNLP	4	14,2547	XPRESS,PATHNLP	4	21,1764
GUROBI,SNOPT	4	14,2547	XPRESS,SNOPT	4	21,2446

Tabla A.20: Tabla de desviaciones medias para cada combinación de niveles de los métodos auxiliares de MILP y NLP correspondiente al algoritmo GAMS/AlphaECP en el caso $C_p = 1,2$, para la experiencia numérica de la subsección 3.5.2.

Fuente de variabilidad	Suma de cuadrados	g.l.	Cuadrado medio	Cociente F	p -valor
Efectos principales					
MILP	0,110234	7	0,0157478	2,84	0,0088
NLP	9,33504	6	1,55584	280,53	0,0000
U. exper.	1,09014	3	0,363379	65,52	0,0000
Interacciones					
MILP · NLP	0,213022	42	0,00507196	0,91	0,6212
MILP · u. exper.	0,375844	21	0,0178974	3,23	0,0000
NLP · u. exper.	1,86627	18	0,103682	18,69	0,0000
Residuos	0,698795	126	0,005546		
Total (corregido)	13,6893	223			

Tabla A.21: ANOVA correspondiente a la calibración del algoritmo GAMS/DICOPT en el caso $C_p = 1,2$, para la experiencia numérica de la subsección 3.5.2.

Nivel	N. obs.	Desv. media (%)	Nivel	N. obs.	Desv. media (%)	Nivel	N. obs.	Desv. media (%)
MILP			NLP			U. exper.		
BDMLP	28	13,3385	CONOPT	32	0	6	56	12,3108
CBC	28	13,3385	IPOPT	32	46,6848	7	56	3,98026
CPLEX	28	13,3385	LGO	32	0,0252635	8	56	20,7431
GUROBI	28	13,3385	MINOS	32	1,91844	9	56	20,8433
MOSEK	28	19,9449	MSNLP	32	46,6848			
SULUM	28	15,7789	PATHNLP	32	1,91844			
XA	28	13,3385	SNOPT	32	4,05377			
XPRESS	28	13,3385						

Tabla A.22: Tabla de desviaciones medias para cada nivel de los métodos auxiliares de MILP y de NLP y cada unidad experimental correspondiente al algoritmo GAMS/DICOPT en el caso $C_p = 1,2$, para la experiencia numérica de la subsección 3.5.2.

Combinación	N. de obs.	Desviación media (%)	Combinación	N. de obs.	Desviación media (%)
BDMLP·CONOPT	4	0	MOSEK·CONOPT	4	0
BDMLP·IPOPT	4	46,6848	MOSEK·IPOPT	4	46,6848
BDMLP·LGO	4	0	MOSEK·LGO	4	0,202108
BDMLP·MINOS	4	0	MOSEK·MINOS	4	15,3476
BDMLP·MSNLP	4	46,6848	MOSEK·MSNLP	4	46,6848
BDMLP·PATHNLP	4	0	MOSEK·PATHNLP	4	15,3476
BDMLP·SNOPT	4	0	MOSEK·SNOPT	4	15,3476
CBC·CONOPT	4	0	SULUM·CONOPT	4	0
CBC·IPOPT	4	46,6848	SULUM·IPOPT	4	46,6848
CBC·LGO	4	0	SULUM·LGO	4	0
CBC·MINOS	4	0	SULUM·MINOS	4	0
CBC·MSNLP	4	46,6848	SULUM·MSNLP	4	46,6848
CBC·PATHNLP	4	0	SULUM·PATHNLP	4	0
CBC·SNOPT	4	0	SULUM·SNOPT	4	17,0826
CPLEX·CONOPT	4	0	XA·CONOPT	4	0
CPLEX·IPOPT	4	46,6848	XA·IPOPT	4	46,6848
CPLEX·LGO	4	0	XA·LGO	4	0
CPLEX·MINOS	4	0	XA·MINOS	4	0
CPLEX·MSNLP	4	46,6848	XA·MSNLP	4	46,6848
CPLEX·PATHNLP	4	0	XA·PATHNLP	4	0
CPLEX·SNOPT	4	0	XA·SNOPT	4	0
GUROBI·CONOPT	4	0	XPRESS·CONOPT	4	0
GUROBI·IPOPT	4	46,6848	XPRESS·IPOPT	4	46,6848
GUROBI·LGO	4	0	XPRESS·LGO	4	0
GUROBI·MINOS	4	0	XPRESS·MINOS	4	0
GUROBI·MSNLP	4	46,6848	XPRESS·MSNLP	4	46,6848
GUROBI·PATHNLP	4	0	XPRESS·PATHNLP	4	0
GUROBI·SNOPT	4	0	XPRESS·SNOPT	4	0

Tabla A.23: Tabla de desviaciones medias para cada combinación de niveles de los métodos auxiliares de MILP y NLP correspondiente al algoritmo GAMS/DICOPT en el caso $C_p = 1,2$, para la experiencia numérica de la subsección 3.5.2.

Fuente de variabilidad	Suma de cuadrados	g.l.	Cuadrado medio	Cociente F	p -valor
Efectos principales					
NLP	0,00625483	6	0,00104247	1,02	0,4412
U. exper.	0,3252	3	0,1084	106,51	0,0000
Residuos	0,0183188	18	0,00101771		
Total (corregido)	0,349773	27			

Tabla A.24: ANOVA correspondiente a la calibración del algoritmo GAMS/OQNLP en el caso $C_p = 1,2$, para la experiencia numérica de la subsección 3.5.2.

Nivel	Núm. de obs.	Desviación media (%)	Nivel	Núm. de obs.	Desviación media (%)
NLP			U. exper.		
CONOPT	4	11,4276	6	7	6,44703
IPOPT	4	15,4437	7	7	13,9309
LGO	4	11,4276	8	7	28,9243
MINOS	4	11,4276	9	7	0
MSNLP	4	13,6975			
PATHNLP	4	11,4276			
SNOPT	4	11,4275			

Tabla A.25: Tabla de desviaciones medias para cada nivel del método auxiliar de NLP y cada unidad experimental correspondiente al algoritmo GAMS/OQNLP en el caso $C_p = 1,2$, para la experiencia numérica de la subsección 3.5.2.

Fuente de variabilidad	Suma de cuadrados	g.l.	Cuadrado medio	Cociente F	p -valor
Efectos principales					
NLP	0,594694	6	0,0991157	8,03	0,0003
U. exper.	0,0388398	3	0,0129466	1,05	0,3950
Residuos	0,222132	18	0,0123407		
Total (corregido)	0,855667	27			

Tabla A.26: ANOVA correspondiente a la calibración del algoritmo GAMS/SBB en el caso $C_p = 1,2$, para la experiencia numérica de la subsección 3.5.2.

Nivel	Núm. de obs.	Desviación media (%)	Nivel	Núm. de obs.	Desviación media (%)
NLP			U. exper.		
CONOPT	4	2,4399	6	7	7,11987
IPOPT	4	2,4399	7	7	3,66817
LGO	4	2,4399	8	7	13,9853
MINOS	4	44,1191	9	7	8,91285
MSNLP	4	2,4397			
PATHNLP	4	2,63244			
SNOPT	4	2,43981			

Tabla A.27: Tabla de desviaciones medias para cada nivel del método auxiliar de NLP y cada unidad experimental correspondiente al algoritmo GAMS/SBB en el caso $C_p = 1,2$, para la experiencia numérica de la subsección 3.5.2.

Fuente de variabilidad	Suma de cuadrados	g.l.	Cuadrado medio	Cociente F	p -valor
Efectos principales					
Algoritmo	1,13272	12	0,0943929	7,7	0,0000
U. exper.	0,136392	3	0,045464	3,71	0,0201
Residuos	0,441478	36	0,0122633		
Total (corregido)	1,71059	51			

Tabla A.28: ANOVA correspondiente a la comparación de los algoritmos considerados en la experiencia numérica de la subsección 3.5.2, para el caso $C_p = 1,2$.

Nivel	Núm. de obs.	Desviación media (%)	Nivel	Núm. de obs.	Desviación media (%)
Algoritmo			U. exper.		
Solver	4	46,6848	6	13	9,76258
AlphaECP	4	3,48273	7	13	8,82541
Bonmin B-BB	4	0,926633	8	13	21,8126
Bonmin B-ECP	4	34,9606	9	13	13,5066
Bonmin B-Hyb	4	21,1356			
Bonmin B-OA	4	4,64683			
Bonmin B-QG	4	18,4337			
Bonmin B-iFP	4	28,8151			
DICOPT	4	0			
KNITRO	4	0,522776			
LINGOGlobal	4	1,7222			
OQNLP	4	11,4275			
SBB	4	2,4399			

Tabla A.29: Tabla de desviaciones medias para cada algoritmo y cada unidad experimental considerados en la experiencia numérica de la subsección 3.5.2, para el caso $C_p = 1,2$.

Bibliografía

- Abhishek K, Leyffer S, Linderoth J (2010a). FilmINT: An Outer Approximation-Based Solver for Convex Mixed-Integer Nonlinear Programs. *INFORMS Journal on Computing*, 22(4):555–567.
- Abhishek K, Leyffer S, Linderoth JT (2010b). Modeling without categorical variables: a mixed-integer nonlinear program for the optimization of thermal insulation systems. *Optimization and Engineering*, 11(2):185–212.
- Adams WP, Sherali HD (1993). Mixed-integer bilinear programming problems. *Mathematical Programming*, 59(1-3):279–305.
- Adjiman CS, Androulakis IP, Floudas CA (1997). Global optimization of MINLP problems in process synthesis and design. *Computers and Chemical Engineering*, 21(Supplement 1):S445–S450.
- Adjiman CS, Schweiger CA, Floudas CA (1998). Nonlinear and Mixed-Integer Optimization in Chemical Process Network Systems. *Network Design: Connectivity and Facilities Location*, 40:429–452.
- AENOR (2005). *Sistemas de gestión de la calidad. Fundamentos y vocabulario (ISO 9000:2005)*. AENOR, Madrid.
- Agresti A (2002). *Categorical Data Analysis*. 2.^aed. John Wiley & Sons, New York.
- Al-Sultan KS, Rahim MA (eds.) (1997). *Optimization in quality control*. Kluwer Academic Publishers, Boston.
- Alemi F (2004). Tukey's control chart. *Quality Management in Healthcare*, 13(4):216–221.

- Alt FB (1985). Multivariate Quality Control. En: Johnson NL, Kotz S (eds.), *Encyclopedia of Statistical Sciences*. Wiley, New York, NY.
- Altunay M, Leyffer S, Linderoth JT, Xie Z (2011). Optimal response to attacks on the open science grid. *Computer Networks*, 55(1):61–73.
- Amin RW, Miller RW (1993). A robustness study of \bar{X} charts with variable sampling intervals. *Journal of Quality Technology*, 25(1):36–44.
- Angira R, Babu BV (2003). Evolutionary Computation for Global Optimization of Non-Linear Chemical Engineering Processes. En: *Proceedings of International Symposium on Process Systems Engineering and Control (ISPSEC '03)*, págs. 87–91. Bombay, Mumbai.
- Aparisi F (1996). Hotelling's T^2 control chart with adaptive sample sizes. *International Journal of Production Research*, 34(10):2853–2862.
- Aparisi F, de Luna MA (2009). Synthetic- \bar{X} control charts optimized for in-control and out-of-control regions. *Computers and Operations Research*, 36(12):3204–3214.
- Babu BV, Angira R (2002). A Differential Evolution Approach for Global Optimization of MINLP Problems. En: *Proceedings of 4th Asia-Pacific Conference on Simulated Evolution and Learning*. Singapore.
- Balaprakash P, Wild SM, Hovland PD (2011). Can search algorithms save large-scale automatic performance tuning? *Procedia Computer Science*, 4:2136–2145.
- Bartelt-Hunt SL, Culver TB, Smith JA, Shawn Matott L, Rabideau AJ (2006). Optimal Design of a Compacted Soil Liner Containing Sorptive Amendments. *Journal of Environmental Engineering*, 132(7):769–776.
- Basso D, Chiarandini M, Salmaso L (2007). Synchronized permutation tests in $I \times J$ replicated designs. *Journal of Statistical Planning and Inference*, 137(8):2564–2578.
- Beale EML (1977). Integer Programming. En: Jacobs DAH (ed.), *The State of The Art in Numerical Analysis*. Academic Press, London.
- Belotti P, Kirches C, Leyffer S, Linderoth J, Luedtke J, Mahajan A (2013). Mixed-Integer Nonlinear Optimization. *Acta Numerica*, 22(1):1–131.

- Belotti P, Lee J, Liberti L, Margot F, Wächter A (2009). Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4-5):597–634.
- Berthold T (2012). RENS — the optimal rounding. Inf. Téc. ZIB-Report 12-17, DFG Research Center Matheon, Zuse Institute Berlin, Berlin. <http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:0296-matheon-11311>.
- Bhothe KR (1988). *World Class Quality. Design of Experiments Made Easier, More Cost Effective than SPC*. 1.^aed. American Management Association, Saranac Lake, NY.
- Biegler LT, Grossmann IE (2004). Retrospective on optimization. *Computers and Chemical Engineering*, 28(8):1169–1192.
- Bienstock D (1996). Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74(2):121–140.
- Bienstock D, Mattia S (2007). Using mixed-integer programming to solve power grid blackout problems. *Discrete Optimization*, 4(1):115–141.
- Bishop CM (1995). *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.
- Bonami P, Biegler LT, Conn AR, Cornuéjols G, Grossmann IE, Laird CD, Lee J, Lodi A, Margot F, Sawaya N, Wächter A (2008). An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204.
- Bonami P, Cornuéjols G, Lodi A, Margot F (2009). A feasibility pump for mixed integer nonlinear programs. *Mathematical Programming*, 19(2):331–352.
- Bonami P, Gonçalves JP (2012). Heuristics for convex mixed integer nonlinear programs. *Computational Optimization and Applications*, 51(2):729–747.
- Bonami P, Kılınç M, Linderoth J (2012). Algorithms and software for convex Mixed Integer Nonlinear Programs. En: Lee J, Leyffer S (eds.), *Mixed Integer Nonlinear Programming*, 1.^aed., págs. 1–39. The IMA Volumes in Mathematics and its Applications, Springer, New York, NY.
- Bonami P, Lee J (2013). BONMIN Users' Manual. <http://www.coin-or.org/Bonmin/Intro.html>.

- Bonami P, Lee J, Leyffer S, Wächter A (2011). More Branch-and-Bound Experiments in Convex Nonlinear Integer Programming. Inf. Téc. Preprint ANL/MCS-P1949-0911, Argonne National Laboratory, Argonne, IL. http://www.optimization-online.org/DB_HTML/2011/09/3191.html.
- Boorstyn RR, Frank H (1977). Large-scale network topological optimization. *IEEE Transactions on Communications*, 25(1):29–47.
- Borchers B, Mitchell JE (1994). An improved branch and bound algorithm for mixed integer nonlinear programs. *Computers and Operations Research*, 21(4):359–367.
- Boukari D, Fiacco AV (1995). Survey of penalty, exact-penalty and multiplier methods from 1968 to 1993. *Optimization*, 32(4):301–334.
- Bourke PD (1991). Detecting a Shift in Fraction Nonconforming Using Run-Length Control Charts with 100% Inspection. *Journal of Quality Technology*, 23(3):225–238.
- Box GEP, Kramer T (1992). Statistical Process Monitoring and Feedback Adjustment: A Discussion. *Technometrics*, 34(3):251–267.
- Boyd S, Vandenberghe L (2004). *Convex Optimization*. Cambridge University Press, Cambridge.
- Bragalli C, D’Ambrosio C, Lee J, Lodi A, Toth P (2006). An MINLP solution method for a water network problem. En: Azar Y, Erlebach T (eds.), *Algorithms – ESA 2006. 4th Annual European Symposium on Algorithms*, págs. 696–707. Springer, Zurich.
- Brent RP (1971). An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4):422–425.
- Brent RP (1973). *Algorithms for Minimization without Derivatives*. Prentice Hall, Englewood Cliffs, NJ.
- Brown Jr NR (1966). Zero Defects the Easy Way with Target Area Control. *Modern Machine Shop*, July:96–100.
- Burden RL, Faires JD (2010). *Numerical Analysis*. 9.ªed. Brooks Cole, Boston, MA.

- Burer S, Letchford AN (2012). Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97–106.
- Bussieck MR, Pruessner A (2003). Mixed-Integer Nonlinear Programming. *SIAG/Optimization Views-and-News*, 14(1):19–22.
- Bussieck MR, Vigerske S (2011). MINLP Solver Software. En: Cochran JJ, Cox Jr LA, Keskinocak P, Kharoufeh JP, Smith JC (eds.), *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons.
- Byrd RH, Nocedal J, Waltz RA (2006). KNITRO: An Integrated Package for Nonlinear Optimization. En: Pillo G, Roma M (eds.), *Large-Scale Nonlinear Optimization*, págs. 35–59. *Nonconvex Optimization and Its Applications*, Springer US, New York, NY.
- Caballero JA, Grossmann IE (2007). Una revisión del estado del arte en Optimización. *Revista Iberoamericana de Automática e Informática Industrial*, 4(1):5–23.
- Camisón C, González T, Cruz S (2006). *Gestión de la Calidad. Conceptos, enfoque, modelos y sistemas*. 1.ªed. Pearson Educación, Madrid.
- Cao YJ, Wu QH (1997). Mechanical design optimization by mixed-variable evolutionary programming. *IEEE International Conference on Evolutionary Computation, 1997*, 1:443–446.
- Cardoso MF, Salcedo RL, de Azevedo S, Barbosa D (1997). A simulated annealing approach to the solution of minlp problems. *Computers and Chemical Engineering*, 21(12):1349–1364.
- Carlyle WM, Montgomery DC, Runger GC (2000a). Optimization Problems and Methods in Quality Control and Improvement. *Journal of Quality Technology*, 32(1):1–17.
- Carlyle WM, Montgomery DC, Runger GC (2000b). Optimization Problems and Methods in Quality Control and Improvement. Response. *Journal of Quality Technology*, 32(1):30–31.
- Carot V, Jabaloyes JM, Carot T (2002). Combined double sampling and variable sampling interval \bar{X} chart. *International Journal of Production Research*, 40(9):2175–2186.

- Carot Alonso V (1998). *Control estadístico de la calidad*. SPUPV-98.4007, Universidad Politécnica de Valencia. Servicio de publicaciones, Valencia.
- Carroll CW (1961). The Created Response Surface Technique for Optimizing Nonlinear, Restrained Systems. *Operations Research*, 9(2):169–184.
- Castillo I, Westerlund J, Emet S, Westerlund T (2005). Optimization of block layout design problems with unequal areas: A comparison of MILP and MINLP optimization methods. *Computers and Chemical Engineering*, 30(1):54–69.
- Celano G, Castagliola P, Trovato E, Fichera S (2011). Shewhart and EWMA t Control Charts for Short Production Runs. *Quality and Reliability Engineering International*, 27(3):313–326.
- Celano G, Fichera S (1999). Multiobjective economic design of an X control chart. *Computers and Industrial Engineering*, 37(1-2):129–132.
- Chamberlayne JW (1997). Pairs Chart. Inf. téc., Philips UK, Durham.
- Champ CW, Rigdon SE (1997). An analysis of the run sum control chart. *Journal of Quality Technology*, 29(4):407–417.
- Chan LY, Lin DKJ, Xie M, Goh TN (2002). Cumulative probability control charts for geometric and exponential process characteristics. *International Journal of Production Research*, 40(1):133–150.
- Chan LY, Xie M, Goh TN (2000). Cumulative quantity control charts for monitoring production processes. *International Journal of Production Research*, 38(2):397–408.
- Chapra SC, Canale RP (2010). *Numerical Methods for Engineers*. 6.^aed. McGraw-Hill Higher Education, Boston, MA.
- Cheng T, Lau M, Duffuaa S (1997). A Concise Review of Lot-by-lot Acceptance Sampling by Attributes. En: Al-Sultan K, Rahim MA (eds.), *Optimization in Quality Control*. Kluwer Academic Publishers, London.
- Cheung BKS, Langevin A, Delmaire H (1997). Coupling Genetic Algorithm with a Grid Search Method to Solve Mixed Integer Nonlinear Programming Problems. *Computers and Mathematics with Applications*, 34(12):13–23.

- Chou CY, Chen CH (2006). Economic design of variable sampling intervals T^2 control charts using genetic algorithms. *Expert Systems with Applications*, 30(2):233–242.
- Chung KJ (1991). A simplified procedure for the economic design of control charts: a unified approach. *Engineering Optimization*, 17(4):313–320.
- Coello Coello CA (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287.
- Conn AR, Scheinberg K, Vicente LN (2009). *Introduction to derivative-free optimization*. MPS-SIAM Book Series on Optimization, Society for Industrial and Applied Mathematics, Philadelphia, PA.
- Cook SA (1971). The complexity of theorem-proving procedures. En: *Proceedings of the third annual ACM symposium on Theory of computing - STOC '71*, págs. 151–158. New York, NY.
- Corsano G, Vecchiotti AR, Montagna JM (2011). Optimal design for sustainable bioethanol supply chain considering detailed plant performance model. *Computers and Chemical Engineering*, 35(8):1384–1398.
- Costa AFB (1994). \bar{X} charts with variable sample size. *Journal of Quality Technology*, 26(3):155–163.
- Costa AFB (1997). \bar{X} chart with variable sample size and sampling intervals. *Journal of Quality Technology*, 29(2):197–204.
- Costa AFB (1999). Joint \bar{X} and R charts with variable sample sizes and sampling intervals. *Journal of Quality Technology*, 31(4):387–397.
- Costa L, Oliveira P (2001). Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems. *Computers and Chemical Engineering*, 25(2):257–266.
- Costa-Montenegro E, González-Castaño FJ, Rodríguez-Hernández PS, Burguillos-Rial JC (2007). Nonlinear Optimization of IEEE 802.11 Mesh Networks. En: Shi Y, van Albada GD, Dongarra J, Sloot PM (eds.), *Computational Science — ICCS 2007 7th International Conference, Part IV*, págs. 466–473. Springer-Verlag, Beijing.

- Cox S, Garside J, Kotsialos A (2013a). Simulation of High Precision Process Control for Set-up Dominant Processes. *Procedia CIRP*, 11:379–384.
- Cox S, Garside JA, Kotsialos A (2012). Set-up Dominated Low Volume Process Capability Estimation Using a Pre-Control Method. En: *Proceedings of the 1st International Through-Life Engineering Services Conference (TESConf 2012)*, págs. 221–226. Cranfield University Press, Shrivenham.
- Cox S, Garside JA, Kotsialos A (2013b). Discrete-Event Simulation of Process Control in Low Volume High Value Industries. En: *Advances in Manufacturing Technology XXVII. Proceedings of the 11th International Conference on Manufacturing Research (ICMR 2013)*, págs. 599–604. Cranfield University Press, Cranfield.
- Crosier R (1988). Multivariate Generalizations of Cumulative Sum Quality Control Schemes. *Technometrics*, 30(3):291–303.
- Crowder SV (1987). A Simple Method for Studying Run-Length Distributions of Exponentially Weighted Moving Average Charts. *Technometrics*, 29(4):401–407.
- Crowder SV (1989). Design of Exponentially Weighted Moving Average Schemes. *Journal of Quality Technology*, 21(2):155–162.
- Cullen CC, Bothe DR (1989). SPC for short production runs. En: *Proceedings of the National Aerospace and Electronic Conference*. Dayton, OH.
- Cytel Software Corporation (2013). *StatXact 10 on-line user manual*. Cytel Software Corporation, Cambridge, MA.
- Dakin R (1965). A tree-search algorithm for mixed integer programming problems. *Computer Journal*, 8(3):250–255.
- D’Ambrosio C, Lodi A (2011). Mixed integer nonlinear programming tools: a practical overview. *4OR A Quarterly Journal of Operations Research*, 9(4):329–349.
- Das TK, Jain V (1997). An economic design model for \bar{X} charts with random sampling policies. *IIE Transactions*, 29(6):507–518.
- Davis RB, Homer A, Woodall WH (1990). Performance of the zone control chart. *Communications in Statistics-Theory and Methods*, 19(5):1581–1587.

- Del Castillo E, Montgomery DC (1994). Short-Run Statistical Process Control: Q-Chart Enhancements and Alternative Methods. *Quality and Reliability Engineering International*, 10(2):87–97.
- Del Castillo E, Montgomery DC (1996). Corrigenda: Short-run statistical process control: Q-Chart enhancements and alternative methods. *Quality and Reliability Engineering International*, 12(3):157.
- Del Castillo E, Semple J (2000). Optimization Problems and Methods in Quality Control and Improvement. Discussion. *Journal of Quality Technology*, 32(1):20–23.
- Deming WE (1986). *Out of the Crisis*. Massachusetts Institute of Technology, Center for Advanced Engineering Study, Cambridge, MA.
- Di Bucchianico A, Mooiweer GW, Mooner EJG (2005). Monitoring Infrequent Failures of High-volume Production Processes. *Quality and Reliability Engineering International*, 21(5):521–528.
- Dimopoulos GG (2007). Mixed-variable engineering optimization based on evolutionary and social metaphors. *Computer Methods in Applied Mechanics and Engineering*, 196(4-6):803–817.
- Donde V, López V, Lesieutre B, Pinar A, Yang C, Meza J (2005). Identification of Severe Multiple Contingencies in Electric Power Networks. En: *Proceedings of the 37th Annual North American Power Symposium*, págs. 59–66. Ames, IA.
- Drud A (2013). *CONOPT*. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/conopt.pdf>.
- Družovec M, Šostar A, Godina A (1998). MPC process control procedure. *Strojniški Vestnik-Journal of Mechanical Engineering*, 44(1-2):29–34.
- Dudding BP, Jennett WJ (1944). Control Chart Technique When Manufacturing to a Specification. Inf. téc., British Standards Institution, London.
- Duran M, Grossmann IE (1986). An outer-approximation algorithm for a class of Mixed-Integer Nonlinear Programs. *Mathematical Programming*, 36:307–339.
- Ebeling CE (1997). *An Introduction to Reliability and Maintainability Engineering*. McGraw-Hill, Boston, MA.

- Edgington ES, Onghena P (2007). *Randomization tests*. 4.^aed. Statistics: A Series of Textbooks and Monographs, CRC Press, Boca Ratón, FL.
- Egea JA, Rodríguez-Fernández M, Banga JR, Martí R (2007). Scatter search for chemical and bio-process optimization. *Journal of Global Optimization*, 37(3):481–503.
- Ehrgott M (2005). *Multicriteria Optimization*. Springer, Berlin.
- Eliceche AM, Corvalán SM, Martínez P (2007). Environmental life cycle impact as a tool for process optimisation of a utility plant. *Computers and Chemical Engineering*, 31(5-6):648–656.
- Elwalid A, Mitra D, Wang Q (2006). Distributed Nonlinear Integer Optimization for Data-Optical Internetworking. *IEEE Journal on Selected Areas in Communications*, 24(8):1502–1513.
- Epperson JF (2007). *An Introduction to Numerical Methods and Analysis*. John Wiley & Sons, Hoboken, NJ.
- Evans JR, Lindsay WM (2008). *Administración y control de la calidad*. 7.^aed. Paraninfo, México.
- Farnum NR (1992). Control Charts for Short Runs: Nonconstant Process and Measurement Error. *Journal of Quality Technology*, 24(2):138–144.
- Ferrer-Riquelme AJ (2009). Statistical Control of Measures and Processes. En: Brown SD, Tauler Ferré R, Walczak B (eds.), *Comprehensive Chemometrics. Chemical and Biochemical Data Analysis*, cap. 1.04, págs. 97–126. Elsevier, Amsterdam.
- Ferris MC (2013). PATHNLP. En: *GAMS — The Solver Manuals*, págs. 415–416. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/pathnlp.pdf>.
- Fiacco AV, McCormick GP (1966). Extensions of SUMT for Nonlinear Programming: Equality Constraints and Extrapolation. *Management Science*, 12(11):816–828.
- FICO (2013). XPRESS. En: *GAMS — The Solver Manuals*, págs. 489–500. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/xpress.pdf>.

- Fisher RA (1925). *Statistical Methods for Research Workers*. Biological monographs and manuals, Oliver and Boyd, Edinburgh, London.
- Fisher RA (1935). *The Design of Experiments*. Oliver and Boyd, Edinburgh, London.
- Fletcher R, Leyffer S (1994). Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66:327–349.
- Floudas CA (1995). *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Topics in chemical engineering, Oxford University Press, New York.
- Floudas CA, Gounaris CE (2009). A review of recent advances in global optimization. *Journal of Global Optimization*, 45(1):3–38.
- Floudas CA, Misener R (2013a). ANTIGONE. En: *GAMS — The Solver Manuals*, págs. 25–38. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/antigone.pdf>.
- Floudas CA, Misener R (2013b). ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. *Submitted for publication*.
- Free Software Foundation, Inc (2009). GSL - GNU Scientific Library. <http://www.gnu.org/software/gsl/>.
- Freire AS, Moreno E, Vielma JP (2012). An integer linear programming approach for bilinear integer programming. *Operations Research Letters*, 40(2):74–77.
- Freund RA (1957). Acceptance control charts. *Industrial Quality Control*, 14(4):13–23.
- Frontline Systems Inc (2013a). Basic Solver - Optimization Methods. <http://www.solver.com/content/basic-solver-optimization-methods>.
- Frontline Systems Inc (2013b). Excel Solver Help. <http://www.solver.com/excel-solver-help>.
- Fügenschuh A, Vigerske S, Homfeld H, Schülldorf H (2010). Mixed-Integer Nonlinear Problems in Transportation Applications. En: *2nd International Conference on Engineering Optimization*. Lisboa.

- Fylstra D, Lasdon LS, Watson J, Waren AD (1998). Design and Use of the Microsoft Excel Solver. *Interfaces*, 28(5):29–55.
- Galassi M, Davies J, Theiler J, Gough B, Jungman G, Alken P, Booth M, Rossi F (2009). *GNU Scientific Library Reference Manual*. 3.^aed. Network Theory, Guildford.
- GAMS Development Corp (2013a). BDMLP. En: *GAMS — The Solver Manuals*, págs. 503–504. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/bdmlp.pdf>.
- GAMS Development Corp (2013b). CPLEX 12. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/cplex.pdf>.
- GAMS Development Corp (2013c). SBB. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/sbb.pdf>.
- GAMS Development Corp (2013d). SCIP. En: *GAMS — The Solver Manuals*, págs. 734–798. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/scip.pdf>.
- García-Díaz JC, Aparisi F (2005). Economic design of EWMA control charts using regions of maximum and minimum ARL. *IIE Transactions*, 37(11):1011–1021.
- Geißler B, Martin A, Morsi A, Schewe L (2012). Using piecewise linear functions for solving MINLPs. En: Lee J, Leyffer S (eds.), *Mixed Integer Nonlinear Programming*, 1.^aed., págs. 287–314. The IMA Volumes in Mathematics and its Applications, Springer, New York, NY.
- Geoffrion AM (1972). Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260.
- Gill PE, Murray W, Saunders MA, Drud A (2013). SNOPT. En: *GAMS — The Solver Manuals*, págs. 425–456. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/snopt.pdf>.
- Glover F (1977). Heuristics for Integer Programming Using Surrogate Constraints. *Decision Sciences*, 8(1):156–166.
- Glover F (1998). A template for scatter search and path relinking. En: Hao JK, Lutton E, Ronald E, Schoenauer M, Snyers D (eds.), *Artificial Evolution: Third*

- European Conference; Selected Papers / AE '97. Nîmes, France, October 22–24, 1997, Lecture Notes in Computer Science*, volumen 1363, págs. 13–54. Springer-Verlag, Berlin.
- Goberna MA, López MA (1998). *Linear Semi-Infinite Optimization*. John Wiley.
- Goh TN (1987). A control chart for very high yield processes. *Quality Assurance*, 13(1):18–22.
- Good P (1993). *Permutation tests*. Springer Verlag, New York.
- Goyal S, Gunasekaran A, Martikainen T, Yli-Olli P (1993). Integrating Production and Quality Control Policies: A survey. *European Journal of Operational Research*, 69(1):1–13.
- Grossmann IE (2002). Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering*, 3(3):227–252.
- Grossmann IE, Sargent RW (1979). Optimum design of multipurpose chemical plants. *Industrial and Engineering Chemistry Process Design and Development*, 18(2):343–348.
- Grossmann IE, Viswanathan J, Vecchiotti A, Raman R, Kalvelagen E (2013). *DICOPT*. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/dicopt.pdf>.
- Gruska GF, Heaphy MS (1991). Stop Light Control - Revisited. *American Society for Quality Control, Statistics Division Newsletter*, 11(4):11–12.
- Guerra A, Newman AM, Leyffer S (2011). Concrete Structure Design using Mixed-Integer Nonlinear Programming with Complementarity Constraints. *SIAM Journal on Optimization*, 21(3).
- Guh RS, Tannock JDT, O'Brien C (1999). IntelliSPC: a hybrid intelligent tool for on-line economical statistical process control. *Expert Systems with Applications*, 17(3):195–212.
- Guo W, Huang X (2011). Achieving capacity fairness for wireless mesh networks. *Wireless Communications and Mobile Computing*, 11(5):632–643.

- Gupta OK, Ravindran A (1981). Nonlinear Integer Programming and Discrete Optimization. En: Mayne RW, Ragsdell KM (eds.), *Progress in Engineering Optimization*, págs. 27–32. American Society of Mechanical Engineers, New York.
- Gupta OK, Ravindran A (1985). Branch and bound experiments in convex nonlinear integer programming. *Management Science*, 31(12):1533–1546.
- Gurobi Optimization Inc (2013). GUROBI 5.5. En: *GAMS — The Solver Manuals*, págs. 199–222. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/gurobi.pdf>.
- Harjunkski I, Westerlund T, Pörn R (1999). Numerical and environmental considerations on a complex industrial mixed integer non-linear programming (MINLP) problem. *Computers and Chemical Engineering*, 23(10):1545–1561.
- Harris B (2007). Lagrange: A Well-Behaved Function. *The Montana Mathematics Enthusiast*, 4(1):128–137.
- Hawkins DM (1987). Self-Starting Cusum Charts for Location and Scale. *The Statistician*, 36(4):299–316.
- Hawkins DM (1993). Cumulative sum control charting: An underutilized SPC tool. *Quality Engineering*, 5(3):463–477.
- Hawkins DM, Olwell DH (1998). *Cumulative sum charts and charting for quality improvement*. 1.^aed. Statistics for Engineering and Physical Science, Springer-Verlag, New York, NY.
- He Q, Wang L (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence*, 20(1):89–99.
- Heaphy MS, Gruska GF (1984). Stop Light Control. *American Society for Quality Control, Statistics Division Newsletter*, 5(2).
- Hill D (1956). Modified Control Limits. *Applied Statistics*, 5(1):12–19.
- Hillier FS, Lieberman GJ (2010). *Introducción a la investigación de operaciones*. 9.^aed. McGraw-Hill, México.

- Hiriart-Urruty JB, Lemaréchal C (2001). *Fundamentals of Convex Analysis*. Grundlehren Text Editions, Springer, Berlin.
- Hoerl RW, Palm AC (1992). Discussion: Integrating SPC and APC. *Technometrics*, 34(3):268–272.
- Holmes DS, Mergen AE (2000). Building an Acceptance Chart. *Quality Digest*, 20(6):35–39.
- Horst R, Tuy H (1996). *Global Optimization. Deterministic Approaches*. 3.ªed. Springer, Berlin.
- Hotelling H (1931). The generalization of Student's ratio. *Annals of Mathematical Statistics*, 2(3):360–378.
- Hotelling H (1947). Multivariate quality control. En: Eisenhart C, Hastay M, Wallis W (eds.), *Techniques of Statistical Analysis*, págs. 111–184. McGraw-Hill, New York.
- Hubele NF (1988). A multivariate and stochastic framework for statistical process control. En: Keats J, Hubele NF (eds.), *Statistical Process Control In Automated Manufacturing*, 1.ªed., cap. 8, págs. 129–151. Marcel Dekker, New York.
- Hunter JS (1986). The exponentially weighted moving average. *Journal of Quality Technology*, 18(4):203–210.
- Ishikawa K (1985). *What is Total Quality Control? The Japanese Way*. Prentice Hall, Englewood Cliffs, NJ.
- Jaehn AH (1987). Zone control charts—SPC made easy. *Quality*, 26:51–53.
- Jaehn AH (1991). The Zone Control Chart. *Quality Progress*, 24(7):65–68.
- Jin C, Davis RB (1991). Calculation of average run lengths for zone control charts with specified zone scores. *Journal of Quality Technology*, 23(4):355–358.
- Jobst N, Horniman M, Lucas C, Mitra G (2001). Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. *Quantitative Finance*, 1(5):489–501.
- Johnson ME (2000). Optimization Problems and Methods in Quality Control and Improvement. Discussion. *Journal of Quality Technology*, 32(1):28–29.

- Jones DR, Schonlau M, Welch WJ (1998). Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492.
- Juran JM, Gryna FM (eds.) (1993). *Manual de Control de Calidad*, volumen 2. 4.^aed. McGraw-Hill, Aravaca, Madrid.
- Karuppiah R, Grossmann IE (2006). Global optimization for the synthesis of integrated water systems in chemical processes. *Computers and Chemical Engineering*, 30(4):650–673.
- Kelley Jr J (1960). The Cutting-Plane Method for Solving Convex Programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712.
- Kesavan P, Barton PI (2000). Generalized branch-and-cut framework for mixed-integer nonlinear optimization problems. *Computers and Chemical Engineering Chemical Engineering*, 24(2-7):1361–1366.
- Kethler RB, Peters MH (2004). Extending economic design of p charts to handle user specified constraints using a genetic algorithm. *IEEE Transactions*, 36(9):855–864.
- Kincaid R (1992). Minimizing distortion and internal forces in truss structures via simulated annealing. *Structural Optimization*, 4(1):55–61.
- Kocis GR, Grossmann IE (1987). Relaxation Strategy for the Structural Optimization of Process Flowsheets. *Industrial and Engineering Chemistry Research*, 26(9):1869–1880.
- Kocis GR, Grossmann IE (1989). Computational Experience with DICOPT solving MINLP Problems in Process Systems Engineering. *Computers and Chemical Engineering*, 13(3):307–315.
- Köppe M (2011). On the complexity of nonlinear mixed-integer optimization. En: Lee J, Leyffer S (eds.), *Mixed-Integer Nonlinear Programming*, págs. 533–558. IMA Volumes in Mathematics and its Applications, Springer, Berlin.
- Krehbiel TC, Eighme JE, Havelka D (2013). Fast Initial Response Control Charts For Accounting Activities. *Journal of Business and Economics Research*, 11(9):381–392.

- Krehbiel TC, Havelka D, Scharfenort M (2007). Process Monitoring In Accounting: Implementing Pre-Control Charts. *Journal of Applied Business Research*, 23(4):93–103.
- Kruger U, Xie L (2012). *Statistical Monitoring of Complex Multivariate Processes With Application in Industrial Process Control*. 1.^aed. Statistics in Practice, John Wiley & Sons, Chichester.
- Laguna M, Martí R (2002). The OptQuest Callable Library. En: Voß S, Woodruff DL (eds.), *Optimization Software Class Libraries*, cap. 7, págs. 193–218. Operations Research/Computer Science Interfaces Series, Kluwer Academic Publishers, Boston, MA.
- Laguna M, Martí R (2003). *Scatter Search: Methodology and Implementations in C*. Kluwer Academic Publishers, Boston, MA.
- Laguna M, Martí R (2005). Experimental Testing of Advanced Scatter Search Designs for Global Optimization of Multimodal Functions. *Journal of Global Optimization*, 33(2):235–255.
- Lasdon LS, Waren AD, Jain A, Ratner M (1978). Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Programming. *ACM Transactions on Mathematical Software*, 4(1):34–50.
- Lastusilta T (2011). *GAMS MINLP Solver Comparisons and Some Improvements to the AlphaECP Algorithm*. Phd thesis, Åbo Akademi University, Åbo. <https://www.doria.fi/handle/10024/72508>.
- Ledolter J, Swersey A (1997). An evaluation of pre-control. *Journal of Quality Technology*, 29(2):163–171.
- Lee HJ, Lim TJ, Jang SC (2007). VSSI \bar{X} Control Charts for Processes With Multiple Assignable Causes. En: *2007 IEEE International Conference on Industrial Engineering and Engineering Management*, págs. 1241–1245. Singapore.
- Lee J, Leyffer S (eds.) (2012). *Mixed Integer Nonlinear Programming*. 1.^aed. The IMA Volumes in Mathematics and its Applications, Springer, New York, NY.
- León T, Sanmatías S, Vercher E (1998). A multi-local optimization algorithm. *Top*, 6(1):1–18.

- Leyffer S (1993). *Deterministic Methods for Mixed Integer Nonlinear Programming*. Phd thesis, University of Dundee, Department of Mathematics & Computer Science, Dundee. <http://www-unix.mcs.anl.gov/~leyffer/papers/thesis.pdf>.
- Leyffer S (2001). Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Computational Optimization and Applications*, 18(3):295–309.
- Leyffer S (2010). Experiments with MINLP Branching Techniques. En: Bonami P, Liberti L, Miller AJ, Sartenaer A (eds.), *Proceedings of the European Workshop on Mixed Integer Nonlinear Programming*, págs. 45–55. Marseille.
- Liberti L, Cafieri S, Tarissan F (2009). Reformulations in Mathematical Programming: A Computational Approach. En: Abraham A, Hassanien AE, Siarry P, Engelbrecht A (eds.), *Foundations of Computational Intelligence, Studies in Computational Intelligence*, volumen 3, págs. 153–234. Springer, Berlin.
- Liberti L, Mladenović N, Nannicini G (2011). A recipe for finding good solutions to MINLPs. *Mathematical Programming Computation*, 3(4):349–390.
- Lin B, Miller DC (2004). Tabu search algorithm for chemical process optimization. *Computers and Chemical Engineering*, 28(11):2287–2306.
- Lin Y, Schrage L (2009). The global solver in the LINDO API. *Optimization Methods and Software*, 24(4-5):657–668.
- Lin YC, Wang FS, Hwang KS (1999). A hybrid method of evolutionary algorithms for mixed-integer nonlinear optimization problems. *Proceedings of the 1999 Congress on Evolutionary Computation, 1999 CEC 99*, 3:2159–2166.
- LINDO Systems Inc (2013a). LINDO/LINDOGlobal. En: *GAMS — The Solver Manuals*, págs. 251–288. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/lindo.pdf>.
- LINDO Systems Inc (2013b). *LINGO. User's guide*. LINDO Systems Inc., Chicago, IL. <http://www.lindo.com/downloads/PDF/LINGO14.PDF>.
- Lowry CA, Woodall WH, Champ CW, Rigdon SE (1992). A Multivariate Exponentially Weighted Moving Average Control Chart. *Technometrics*, 31(2):189–206.

- Lucas JM, Saccucci MS (1990). Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements. *Technometrics*, 32(1):1–29.
- Lunati Co (2008). Lunati Power. <http://www.lunatipower.com>.
- Luo YQ, Yuan XG, Liu YJ (2007). An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints. *Computers and Chemical Engineering*, 31(3):153–162.
- Luyben ML, Floudas CA (1994). Analyzing the interaction of design and control—1. A multiobjective framework and application to binary distillation synthesis. *Computers and Chemical Engineering*, 18(10):933–969.
- Mackertich NA (1990). Precontrol vs. control charting: a critical comparison. *Quality Engineering*, 2(3):253–260.
- Mahajan A, Leyffer S, Kirches C (2012). Solving Mixed-Integer Nonlinear Programs by QP-Diving. Inf. Téc. Preprint ANL/MCS-P2071-0312, Argonne National Laboratory, Argonne, IL.
- Maria J, Truong TT, Yao J, Lee TW, Nuzzo RG, Leyffer S, Gray SK, Rogers JA (2009). Optimization of 3D Plasmonic Crystal Structures for Refractive Index Sensing. *Journal of Physical Chemistry C*, 113(24):10493–10499.
- Maroto Álvarez MC, Alcaraz Soria J, Ruiz García R (2002). *Investigación operativa: modelos y técnicas de optimización*. Editorial UPV, Valencia.
- Martí R (2006). Scatter Search—Wellsprings and Challenges. *European Journal of Operational Research*, 169(2):351–358.
- Martin A, Möller M, Moritz S (2006). Mixed Integer Models for the Stationary Case of Gas Network Optimization. *Mathematical Programming, Series B*, 105(2-3):563–582.
- Martin T (1993). Shewhart charts & pre-control: rivals or teammates? *American Society for Quality Control, Statistics Division Newsletter*, 13(3):10–13.
- Mathews JH, Fink KK (2004). *Numerical Methods Using Matlab*. 4.^aed. Prentice-Hall, Upper Saddle River, NJ.

- McCarl BA, Meeraus A, van der Eijk P, Bussieck MR, Dirkse S, Steacy P, Nelissen F (2013). *McCarl GAMS User Guide*. GAMS Development Corporation, Washington, DC. http://www.gams.com/dd/docs/bigdocs/gams2002/mccarlgamsuserguide_web.pdf.
- McCormick GP (1976). Computability of global solutions to factorable non-convex programs: Part I — Convex underestimating problems. *Mathematical Programming*, 10(1):147–175.
- Mehta CP (1991). StatXact: A Statistical Package for Exact Nonparametric Inference. *The American Statistician*, 45(1):74–75.
- Meyer CA, Floudas CA, Neumaier A (2002). Global Optimization with Nonfactorable Constraints. *Industrial and Engineering Chemistry Research*, 41(25):6413–6424.
- Michalewicz Z, Fogel DB (2000). *How to solve it: Modern Heuristics*. 2.^ªed. Springer, Berlin.
- Miettinen K (1999). *Nonlinear Multiobjective Optimization*. Kluwer, Massachusetts.
- Minne L, Eslami S, de Keizer N, de Jonge E, de Rooij SE, Abu-Hanna A (2012a). Effect of changes over time in the performance of a customized SAPS-II model on the quality of care assessment. *Intensive Care Medicine*, 38(1):40–46.
- Minne L, Eslami S, de Keizer N, de Jonge E, de Rooij SE, Abu-Hanna A (2012b). Statistical Process Control for Monitoring Standardized Mortality Ratios of a Classification Tree Model. *Methods of Information in Medicine*, 51(4):353–358.
- Misener R, Floudas CA (2009). A review of advances for the pooling problem: modeling, global optimization, and computational studies. *Applied and Computational Mathematics*, 8(1):3–22.
- Misener R, Floudas CA (2012). Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. *Mathematical Programming, Series B*, 136(1):155–182.
- Misener R, Floudas CA (2013). GloMIQO: Global mixed-integer quadratic optimizer. *Journal of Global Optimization*, 57(1):3–50.

- Montgomery DC (2012). *Introduction to Statistical Quality Control*. 7.^aed. John Wiley & Sons, Hoboken, NJ.
- Montgomery DC (2013). *Design and analysis of experiments*. 8.^aed. John Wiley & Sons, Hoboken, NJ.
- MOSEK ApS (2013). MOSEK. En: *GAMS — The Solver Manuals*, págs. 319–368. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/mosek.pdf>.
- Munawar S (2005). A Nonlinear Transformation Based Hybrid Evolutionary Method For MINLP Solution. *Chemical Engineering Research and Design*, 83(A10):1218–1236.
- Murray W, Shanbhag UV (2006). A Local Relaxation Approach for the Siting of Electrical Substations. *Computational Optimization and Applications*, 33(1):7–49.
- Murtagh BA, Saunders MA, Murray W, Gill PE (2013). MINOS. En: *GAMS — The Solver Manuals*, págs. 289–318. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/minos.pdf>.
- Nannicini G, Belotti P (2012). Rounding-based heuristics for nonconvex MINLPs. *Mathematical Programming Computation*, 4(1):1–31.
- Nannicini G, Belotti P, Liberti L (2008). A Local Branching Heuristic for MINLPs. Inf. Téc. 0812.2188, arXiv.org, Cornell University Library. <http://arxiv.org/abs/0812.2188>.
- Navidi WC (2011). *Statistics for Engineers and Scientists*. 3.^aed. McGraw-Hill, New York, NY.
- Nelson LS (1999). Notes on the Shewhart Control Chart. *Journal of Quality Technology*, 31(1):124–126.
- Nelson PR, Wiecek MM (2000). Optimization Problems and Methods in Quality Control and Improvement. Discussion. *Journal of Quality Technology*, 32(1):24–27.
- Olorunniwo F, Knight JE (1995). Monitoring process parameters with pre-control. *Journal of Operations Management*, 12(2):75–87.

- Optimal Methods Inc, OptTek Systems Inc (2013). OQNLP and MSNLP. En: *GAMS — The Solver Manuals*, págs. 369–380. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/oqnlp.pdf>.
- Ott ER, Mundel AB (1954). Narrow Limit Gaging. *Industrial Quality Control*, 10(5):2–9.
- Owen AB (2001). *Empirical Likelihood*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Chapman & Hall/CRC, Boca Raton, FL.
- Page ES (1954). Continuous Inspection Schemes. *Biometrics*, 41(1-2):100–115.
- Page ES (1961). Cumulative Sum Charts. *Technometrics*, 3(1):1–9.
- Pan JN (2007). A study of multivariate pre-control charts. *International Journal of Production Economics*, 105(1):160–170.
- Pardalos PM, Chaovalitwongse W, Iasemidis LD, Sackellares JC, Shiau DS, Carney PR, Prokopyev OA, Yatsenko VA (2004). Seizure warning algorithm based on optimization and nonlinear dynamics. *Mathematical Programming, Series B*, 101(2):365–385.
- Pignatiello Jr JJ, Runger GC (1990). Comparison of Multivariate CUSUM Charts. *Journal of Quality Technology*, 22(3):173–186.
- Pillet M (1996). A Specific SPC Chart For Small-Batch Control. *Quality Engineering*, 8(4):581–586.
- Pintér JD (2013). LGO. En: *GAMS — The Solver Manuals*, págs. 245–250. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/lgo.pdf>.
- Prabhu SS, Montgomery DC, Runger GC (1994). A combined adaptive sample size and sampling interval \bar{X} control scheme. *Journal of Quality Technology*, 26(3):164–176.
- Prabhu SS, Montgomery DC, Runger GC (1997). Economic-statistical design of an adaptive \bar{X} chart. *International Journal of Production Economics*, 49(1):1–15.
- Prabhu SS, Runger GC, Keats JB (1993). \bar{X} chart with adaptive sample sizes. *International Journal of Production Research*, 31(12):2895–2909.

- Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007). *Numerical Recipes. The Art Of Scientific Computing*. 3.^aed. Cambridge University Press, Cambridge.
- Quaglino MB (2003). *Un gráfico para el control de procesos de alto rendimiento basado en la binomial negativa*. Tesis Doctoral, Universitat Politècnica de València.
- Quesada I, Grossmann IE (1992). An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems. *Computers and Chemical Engineering*, 16(10-11):937–947.
- Quesenberry C (1991a). SPC Q Charts for a Binomial Parameter p : Short or Long Runs. *Journal of Quality Technology*, 23(3):239–246.
- Quesenberry C (1991b). SPC Q Charts for a Poisson Parameter λ : Short or Long Runs. *Journal of Quality Technology*, 23(4):296–303.
- Quesenberry C (1991c). SPC Q Charts for Start-Up Processes and Short or Long Runs. *Journal of Quality Technology*, 23(3):213–224.
- Quesenberry C (1995a). Geometric Q Charts for High Quality Processes. *Journal of Quality Technology*, 27(4):304–315.
- Quesenberry C (1995b). On Properties of Binomial Q-charts for Attributes. *Journal of Quality Technology*, 27(3):204–213.
- Quesenberry C (1995c). On Properties of Poisson Q Charts for Attributes. *Journal of Quality Technology*, 27(4):293–303.
- Quesenberry C (1995d). On Properties of Q Charts for Variables. *Journal of Quality Technology*, 27(3):184–203.
- Quist A, van Geemert R, Hoogenboom J, Illés T, Roos C, Terlaky T (1999). Application of nonlinear optimization to reactor core fuel reloading. *Annals of Nuclear Energy*, 26(5):423–448.
- Rahim MA, Al-Sultan KS (1997). Some Contemporary Approaches to Optimization Models in Process Control. En: Al-Sultan KS, Rahim MA (eds.), *Optimization in Quality Control*. Kluwer Academic Publishers, London.
- Rahim MA, Ben-Daya M (1996). A Generalized Economic Model for Joint Determination of Production Run, Inspection Schedule, and Control Chart Design. Inf. Téc. 96-032, Faculty of Administration, University of New Brunswick, Fredericton.

- Rahim MA, Ohta H, Ben-Daya M (1996). An Integrated Dynamic Joint Optimization Model for Controlling both Quality and Maintenance Policies. Inf. Téc. 96-042, Faculty of Administration, University of New Brunswick, Fredericton.
- Rajeev S, Krishnamoorthy C (1992). Discrete Optimization of Structures Using Genetic Algorithms. *Journal of Structural Engineering*, 118(5):1233–1250.
- Rasch D, Guiard V (2004). The robustness of parametric statistical methods. *Psychology Science*, 46(2):175–208.
- Reynolds JH (1971). The run sum control chart procedure. *Journal of Quality Technology*, 3(1):23–27.
- Reynolds MR (1996a). Shewhart and EWMA variable sampling interval control charts with sampling at fixed times. *Journal of Quality Technology*, 28(2):199–212.
- Reynolds MR (1996b). Variable-Sampling-Interval control charts with sampling at fixed times. *IIE transactions*, 28(6):497–510.
- Reynolds MR, Amin RW, Arnold JC, Nachlas JA (1988). \bar{X} charts with variable sampling intervals. *Technometrics*, 30(2):181–192.
- Reynolds MR, Arnold JC, Baik JW (1996). Variable sampling interval \bar{X} charts in the presence of correlation. *Journal of Quality Technology*, 28(1):12–30.
- Ridders C (1979). A new algorithm for computing a single root of a real continuous function. *IEEE Transactions on Circuits and Systems*, 26(11):979–980.
- Ridge E, Kudenko D (2010). Tuning an algorithm using design of experiments. En: Bartz-Beielstein T, Chiarandini M, Paquete L, Preuss M (eds.), *Experimental Methods for the Analysis of Optimization Algorithms*, cap. 11, págs. 265–286. Springer, New York.
- Roberts SW (1959). Control Chart Tests Based on Geometric Moving Averages. *Technometrics*, 1(3):239–250.
- Roberts SW (1966). A Comparison of Some Control Chart Procedures. *Technometrics*, 8(3):411–430.
- Romero Villafranca R, Zúñica Ramajo LR (2008). *Apuntes de control estadístico de calidad*. Editorial UPV, Valencia.

- Rosenthal RE (2013). *GAMS. A User's Guide*. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/bigdocs/GAMSUsersGuide.pdf>.
- Ruiz-Maya Pérez L (1983). *Métodos estadísticos de investigación. Introducción al análisis de la varianza*. 2.^aed. Instituto Nacional de Estadística, Madrid.
- Runger GC, Montgomery DC (1993). Adaptive sampling enhancements for Shewhart control charts. *IIE Transactions*, 25(3):41–51.
- Ryoo HS, Sahinidis NV (1996). A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, 8(2):107–138.
- Sahinidis NV (1996). BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8(2):201–205.
- Sahinidis NV (2013). BARON. En: *GAMS — The Solver Manuals*, págs. 39–54. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/baron.pdf>.
- Salvia AA (1988). Stoplight Control. *Quality Progress*, 21(9):39–42.
- San Matías S, Jabaloyes J, Carrión A (2004). Some modifications of the Classical Pre-control technique. *Quality and Reliability Engineering International*, 20(1):47–60.
- Sandgren E (1990). Nonlinear Integer and Discrete Programming in Mechanical Design Optimization. *Transactions of the ASME, Journal of Mechanical Design*, 112(2):223–229.
- Saniga EM (2000). Optimization Problems and Methods in Quality Control and Improvement. Discussion. *Journal of Quality Technology*, 32(1):18–19.
- Sanmatías S, Vercher E (1998). A Generalized Conjugate Gradient Algorithm. *Journal of Optimization Theory and Applications*, 98(2):489–502.
- Satterthwaite FE (1954). A Simple, Effective Process Control Method. Report 54-1, Rath and Strong, Boston, MA.
- Savelsbergh M (1994). Preprocessing and probing techniques for mixed integer programming problems. *INFORMS Journal on Computing*, 6(4):445–454.

- Shainin D, Shainin P (1989). Pre-Control versus \bar{X} & R charting: continuous or immediate quality improvement? *Quality Engineering*, 1(4):419–429.
- Shewhart WA (1931). *Economic control of quality of manufactured product*. Van Nostrand, New York.
- Soleymani F, Sharifi M (2009). A New Derivative-Free Quasi-Secant Algorithm For Solving Non-Linear Equations. *International Journal of Mathematical and Statistical Sciences*, 1(4):208–210.
- Steiner SH (1997). Pre-control and Some Simple Alternatives. *Quality Engineering*, 10(1):65–74.
- Still C, Westerlund T (2006). Solving Convex MINLP Optimization Problems Using a Sequential Cutting Plane Algorithm. *Computational Optimization and Applications*, 34(1):63–83.
- Still C, Westerlund T (2009). Extended Cutting Plane Algorithm. En: Floudas CA, Pardalos PM (eds.), *Encyclopedia of Optimization*, 2.^aed., págs. 959–966. Springer, New York, NY.
- Stubbs RA, Mehrotra S (1999). A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86(3):515–532.
- Sugg DJ, Miller D (1992). Statistical Process Control For Reel-To-Reel Plating: Thickness Control on Short Runs. En: *The Proceedings of the 79th AESF Annual Technical Conference SURF/FIN '92*, págs. 27–36. American Electroplaters and Surface Finishers Society, Inc., Atlanta, GA.
- Sulum Optimization (2013). SULUM 2.0. En: *GAMS — The Solver Manuals*, págs. 457–480. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/sulum.pdf>.
- Sunset Software Technology (2013). XA. En: *GAMS — The Solver Manuals*, págs. 481–488. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/xa.pdf>.
- Tagaras G (1997). Economic Decision of Time-Varying and Adaptive Control Charts. En: Al-Sultan KS, Rahim MA (eds.), *Optimization in Quality Control*. Kluwer Academic Publishers, London.

- Tagaras G (1998). A Survey of Recent Developments in the Design of Adaptive Control Charts. *Journal of Quality Technology*, 30(3):212–231.
- Taguchi G (1981). *On-line Quality Control During Production*. Japanese Standard Association, Tokyo, Japan.
- Tang PF, Barnett N (1994). A comparison of mean and range charts with Pre-control having particular reference to short-run production. *Quality and Reliability Engineering International*, 10(6):477–485.
- Tawarmalani M, Sahinidis N (2004). Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming, Series A*, 99(3):563–591.
- The COIN-OR Foundation Inc (2012). COIN-OR. <http://www.coin-or.org/>.
- Thomas Jr GB, Weir MD, Hass JR (2014). *Thomas' Calculus*. 13.^{ed}. Pearson.
- Topalidou E, Psarakis S (2009). Review of Multinomial and Multiattribute Quality Control Charts. *Quality and Reliability Engineering International*, 25(7):773–804.
- Torbeck L (2005). The sector chart: A new engineering graph for pharmaceutical processes. *Pharmaceutical technology*, 29(4):102–108.
- Torng CC, Liao HN, Lee PH, Wu JC (2009). Performance evaluation of a Tukey's control chart in monitoring gamma distribution and short run processes. En: *Proceedings of the International MultiConference of Engineers and Computer Scientists*. Hong Kong.
- Traver RW (1985). Pre-Control - A Good Alternative to \bar{X} -R Charts. *Quality Progress*, 18(9):11–14.
- Trietsch D (1999). *Statistical Quality Control: A Loss Minimization Approach, Series on Applied Mathematics*, volumen 10. World Scientific Publishing, Singapore; River Edge, NJ.
- Tukey JW (1949). Comparing Individual Means in the Analysis of Variance. *Biometrics*, 5(2):99–114.
- Ugray Z, Lasdon L, Plummer J, Glover F, Kelly J, Martí R (2005). A multistart scatter search heuristic for smooth NLP and MINLP problems. En: Sharda R, Voß S, Rego C, Alidaee B (eds.), *Metaheuristic Optimization via Memory*

- and Evolution : Tabu Search and Scatter Search, Operations Research/Computer Science Interfaces Series*, volumen 30, págs. 25–57. Kluwer Academic Publishers, Dordrecht.
- van de Braak G, Bünnner MJ, Schittkowski K (2004). Optimal Design of Electronic Components by Mixed-Integer Nonlinear Programming. *Optimization and Engineering*, 5:271–294.
- van der Briel MHL, Villalobos JR, Hogg GL, Lindemann T, Mulé AV (2005). America West Airlines Develops Efficient Boarding Strategies. *Interfaces*, 35(3):191–201.
- Van Roy TJ (1983). Cross decomposition for mixed integer programming. *Mathematical Programming*, 25(1):46–63.
- Vermani SK (2000). SPC Modified With Percent Tolerance Precontrol Charts. *Quality Progress*, October:43–48.
- Vielma JP, Ahmed S, Nemhauser G (2010). Mixed-Integer Models for Nonseparable Piecewise-Linear Optimization: Unifying Framework and Extensions. *Operations Research*, 58(2):303–315.
- Vigerske S (2013). COIN-OR. En: *GAMS — The Solver Manuals*, págs. 517–592. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/coin.pdf>.
- Viswanathan J, Grossmann IE (1990). A combined Penalty Function and Outer Approximation Method for MINLP Optimization. *Computers and Chemical Engineering*, 14(7):769–782.
- Vommi VB, Seetala MSN (2007). A new approach to robust economic design of control charts. *Applied Soft Computing*, 7(1):211–228.
- Wah BW, Chen Y (2003). Hybrid Constrained Simulated Annealing and Genetic Algorithms for Nonlinear Constrained Optimization. *International Journal of Computational Intelligence and Applications*, 3(4):331–355.
- Wang Y, Achenie LEK (2002). A hybrid global optimization approach for solvent design. *Computers and Chemical Engineering*, 26(10):1415–1425.

- Westerlund T, Lastusilta T (2013). AlphaECP. En: *GAMS — The Solver Manuals*, págs. 13–24. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/alphaecp.pdf>.
- Westerlund T, Pettersson F (1995). An extended cutting plane method for solving convex MINLP problems. *Computers and Chemical Engineering*, 19(Supplement 1):S131–S136.
- Westerlund T, Pörn R (2002). Solving Pseudo-Convex Mixed Integer Optimization Problems by Cutting Plane Techniques. *Optimization and Engineering*, 3(3):253–280.
- Western Electric Company (1956). *Statistical Quality Control Handbook*. Mack Printing Company, Easton, PA.
- Wolfram S (2014). Wolfram Mathematica 9 Documentation Center. <http://reference.wolfram.com/>.
- Woodall WH (1997). Control Charts Based on Attribute Data: Bibliography and Review. *Journal of Quality Technology*, 29(2):172–183.
- Woodall WH (2000). Controversies and contradictions in statistical process control. *Journal of Quality Technology*, 32(4):341–350.
- Woods RF (1976). Effective, Economic Quality Through the Use of Acceptance Control Charts. *Journal of Quality Technology*, 8(2):81–88.
- Xie M, Goh TN, Ranjan P (2002). Some effective control chart procedures for reliability monitoring. *Reliability Engineering and System Safety*, 77(2):143–150.
- Xiong Y, Rao SS (2004). Fuzzy nonlinear programming for mixed-discrete design optimization through hybrid genetic algorithm. *Fuzzy Sets and Systems*, 146(2):167–186.
- Yee TF, Grossmann IE (1990). Simultaneous optimization models for heat integration—II. Heat exchanger network synthesis. *Computers and Chemical Engineering*, 14(10):1165–1184.
- You F, Leyffer S (2011). Mixed-integer dynamic optimization for oil-spill response planning with integration of a dynamic oil weathering model. *AIChE Journal*, 57(12):3555–3564.

- Young CM (2010). *Modeling and optimization of container inspection systems*. Phd thesis, The State University of New Jersey. <http://hdl.rutgers.edu/1782.1/rucore10001600001.ETD.000056869>.
- Young CT, Zheng Y, Yeh CW, Jang SS (2007). Information-guided genetic algorithm approach to the solution of MINLP problems. *Industrial and Engineering Chemistry Research*, 46(5):1527–1537.
- Ypma TJ (1995). Historical Development of the Newton-Raphson Method. *SIAM Review*, 37(4):531–551.
- Yuan X, Zhang S, Pibouleau L, Domenech S (1988). Une méthode d'optimisation non linéaire en variables mixtes pour la conception de procédés. *Revue Française d'Automatique, d'Informatique et de Recherche Opérationnelle - Recherche Opérationnelle*, 22(4):331–346.
- Zhang CW, Xie M, Goh TN (2005). Economic design of exponential charts for time between events monitoring. *International Journal of Production Research*, 43(23):5019–5032.
- Zhang CW, Xie M, Goh TN (2008). Economic design of cumulative count of conforming charts under inspection by samples. *International Journal of Production Economics*, 111(1):93–104.
- Zhang L, Chen G, Castagliola P (2009). On t and EWMA t charts for monitoring changes in the process mean. *Quality and Reliability Engineering International*, 25(8):933–945.
- Zhou M, Zhao C (2002). An optimization model and multiple matching heuristics for quality planning in manufacturing systems. *Computers and Industrial Engineering*, 42(1):91–101.
- Zhu Y, Kuno T (2003). Global Optimization of Nonconvex MINLP by a Hybrid Branch-and-Bound and Revised General Benders. *Industrial and Engineering Chemical Research*, 42(3):528–539.
- Zhu Y, Kuno T (2006). A Disjunctive Cutting-Plane-Based Branch-and-Cut Algorithm for 0 - 1 Mixed-Integer Convex Nonlinear Programs. *Industrial and Engineering Chemistry Research*, 45(1):187–196.

- Ziena Optimization LLC (2013a). KNITRO. En: *GAMS — The Solver Manuals*, págs. 223–244. GAMS Development Corporation, Washington, DC. <http://www.gams.com/dd/docs/solvers/knitro.pdf>.
- Ziena Optimization LLC (2013b). Ziena Products: KNITRO. <http://ziena.com/knitro.htm>.
- Zimmer LS, Montgomery DC, Runger GC (2000). Guidelines for the application of adaptive control charting schemes. *International Journal of Production Research*, 38(9):1977–1992.
- Zorriassatine F, Tannock JDT (1998). A review of neural networks for statistical process control. *Journal of Intelligent Manufacturing*, 9(3):209–224.
- Zúnica Ramajo LR, Romero Villafranca R, Guilloto de Barrasa F (1997). Un nuevo gráfico para el control de unidades defectuosas. En: *XXIII Congreso Nacional de Estadística e Investigación Operativa*. Valencia.

When Great Trees Fall

*When great trees fall,
rocks on distant hills shudder,
lions hunker down
in tall grasses,
and even elephants
lumber after safety.*

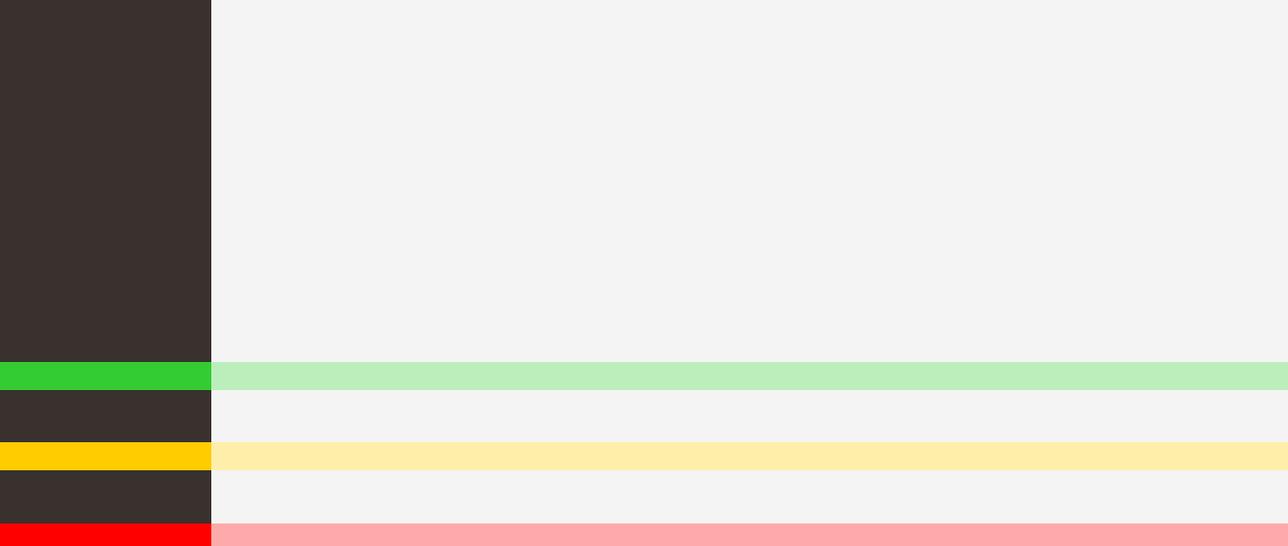
*When great trees fall
in forests,
small things recoil into silence,
their senses
eroded beyond fear.*

*When great souls die,
the air around us becomes
light, rare, sterile.
We breathe, briefly.
Our eyes, briefly,
see with
a hurtful clarity.
Our memory, suddenly sharpened,
examines,
gnaws on kind words
unsaid,
promised walks
never taken.*

*Great souls die and
our reality, bound to
them, takes leave of us.
Our souls,
dependent upon their
nurture,
now shrink, wizened.
Our minds, formed
and informed by their
radiance,
fall away.
We are not so much maddened
as reduced to the unutterable igno-
rance
of dark, cold
caves.*

*And when great souls die,
after a period peace blooms,
slowly and always
irregularly. Spaces fill
with a kind of
soothing electric vibration.
Our senses, restored, never
to be the same, whisper to us.
They existed. They existed.
We can be. Be and be
better. For they existed.*

—Maya Angelou (1928–2014)



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA