



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

---

# Desarrollo de una metodología de optimización multiobjetivo considerando soluciones casi- óptimas. Aplicación a problemas en ingeniería de control

---

Tesis Doctoral

Autor: Alberto Pajares Ferrando

Directores: Dr. Francesc Xavier Blasco Ferragud  
Dr. Juan Manuel Herrero Durá

Valencia, Septiembre 2019



# Agradecimientos

Me gustaría agradecer a todas las personas que de una forma u otra han ayudado durante estos últimos años. En primer lugar, gracias a mis directores Xavi y Juanma, por su dirección que ha llevado por buen camino esta tesis. A Xavi y Gil por introducirme en el mundo multiobjetivo, en mi etapa como becario en la que decidí realizar la tesis. A Juanma, por la gran cantidad de horas dedicadas en este trabajo, además de esos partidos de pádel que permitían desconectar de la rutina. Como no, agradecer también al resto de personas del grupo: Santi, Javi, Raúl, Sergio, Carlos, por hacer los días más amenos con los almuerzos diarios. También quiero agradecer a los compañeros de la Sala: Yadora, Alejandro, Vanessa, Jesús, Frank, Henry, Eslam, Manuel, Julio, Victor, Rubén, Temo, por todas las cervezas y momentos compartidos, sobre todo, durante la etapa inicial de la tesis. También quiero dar una mención especial a mi novia Mori, gracias a su apoyo, ayuda y comprensión todo ha sido más fácil. Gracias a mis hijos Lucas y Vega, por darme el cariño incondicional que me alegra los días. Finalmente, pero no menos importante, gracias a mis padres y hermanos, sin su educación y ejemplo esto no podría haber sido posible.

Este trabajo ha sido parcialmente subvencionado por el Ministerio de Economía y Competitividad a través de la beca FPU15/01652, y los proyectos DPI2015-71443-R y RTI2018-096904-B-I00, por la administración local Generalitat Valenciana a través de la beca ACIF/2015/079 y los proyectos GV/2017/029 y AICO/2019/055, y por la Universitat Politècnica de València a través de la beca FPI-2014/2429.



# Resumen

En un problema de optimización multiobjetivo, habitualmente se busca caracterizar el conjunto de soluciones óptimas de Pareto, ignorando las soluciones casi-óptimas. Sin embargo, estas soluciones pueden proporcionar al diseñador una mayor diversidad de soluciones potencialmente útiles, lo que permite tomar una decisión final más informada. Pese a ello, obtener todas las soluciones casi-óptimas puede aumentar en exceso el número de ellas y, en consecuencia, ralentizar en exceso el proceso de optimización y complicar la etapa de decisión. Por ello, se propone obtener las soluciones casi-óptimas que mayor información relevante aporten al diseñador, descartando el resto de ellas. En este trabajo se asume que las soluciones más relevantes son, además de las óptimas (en el espacio de objetivos), las alternativas casi-óptimas significativamente diferentes (no vecinas en el espacio de parámetros) a las soluciones que le dominan, es decir, las soluciones casi-óptimas no dominadas en su vecindad. Este conjunto de soluciones proporciona alternativas diferentes sin aumentar en exceso el número de ellas. Para caracterizar este conjunto, en esta tesis, se presenta y valida un nuevo algoritmo (nevMOGA). Gracias a este algoritmo y la metodología descrita para su aplicación, el diseñador puede obtener estas soluciones con el objetivo de realizar un análisis más profundo, tomando la decisión final con mayor información. Además, en la tesis, se aplica esta nueva metodología en problemas de identificación de modelos y diseño de controladores multivariables. En ellos, se pone de manifiesto la utilidad de obtener las alternativas casi-óptimas no dominadas en su vecindad, proporcionando nueva información relevante para el diseñador. De hecho, en algunos de estos problemas, las alternativas casi-óptimas son preferidas en lugar de las óptimas.



# Abstract

In a multiobjective optimization problem, the aim is usually to characterize the set of optimal solutions (Pareto set) and the nearly optimal solutions are ignored. Proceeding in this way has a drawback, namely, some of these nearly optimal solutions are potentially useful for the designer and their consideration can lead him or her to make a better informed decision. However, finding all the nearly optimal solutions would excessively slow down the optimization process and would complicate the decision stage unnecessarily. In order to overcome this problem, we propose a new methodology to obtain only the nearly optimal solutions that really provide relevant information to the designer, discarding the rest of them. In this work, it is assumed that the most relevant solutions are, apart from the optimal ones, the nearly optimal solutions which are significantly different (not neighbors in the parameter space) from the solutions that dominate them, that is to say, the nearly optimal solutions non dominated in their neighborhood. In this way, a set of potentially useful alternatives is provided, without increasing their number unnecessarily. In order to characterize this new set of solutions, a novel algorithm (nevMOGA) is presented and validated. Thanks to this algorithm and to the methodology described for its application, a designer will be able to obtain these new solutions and, therefore, this will enable him or her to perform a deeper analysis of the problem, which eventually will result in a more knowledgeable decision. In addition, this new methodology is applied to several engineering problems in the areas of model tuning and multivariable control design. Through these application examples, the usefulness of obtaining and taking into account the nearly optimal solutions non dominated in their neighborhood is demonstrated. In effect, in some of these cases, a nearly optimal solution is preferred to any of the optimal ones.



# Resum

En un problema d'optimització multiobjectiu, habitualment se busca caracteritzar el conjunt de solucions òptimes de Pareto, ignorant les solucions quasi-òptimes. Aquestes solucions poden proporcionar al dissenyador una major diversitat de solucions potencialment útils, la qual cosa permet prendre una decisió final més informada. No obstant això, obtenir totes les solucions quasi-òptimes pot augmentar en excés el número d'elles, alentint en excés el procés d'optimització i complicant l'etapa de decisió. Per això, es proposa obtenir les solucions quasi-òptimes que major informació rellevant aporten al dissenyador, descartant la resta d'elles. En aquest treball s'assumix que les solucions més rellevants són, a més de les òptimes (en l'espai d'objectius), les alternatives quasi-òptimes significativament diferents (no veïnes en l'espai de paràmetres) a les solucions que li dominen, és a dir, les solucions quasi-òptimes no dominades en el seu veïnatge. Aquest conjunt de solucions proporciona alternatives diferents sense augmentar en excés el número d'elles. Per a caracteritzar aquest conjunt, en aquesta tesi, es presenta i valida un nou algorisme (nevMOGA). Gràcies a aquest algorisme i la metodologia descrita per a la seua aplicació, el dissenyador pot obtenir aquestes solucions amb l'objectiu de realitzar una anàlisi més profunda, prenent la decisió final amb major informació. A més, en la tesi, s'aplica aquesta nova metodologia en problemes d'identificació de models i disseny de controladors multivariables. En ells, es posa de manifest la utilitat d'obtenir les alternatives quasi-òptimes no dominades en el seu veïnatge, proporcionant nova informació rellevant per al dissenyador. De fet, en diversos casos, les alternatives quasi-òptimes són preferides en lloc de les òptimes.



# Índice general

Resumen	III
1 Motivación y objetivo de la tesis	1
1.1 Introducción	3
1.2 Motivación	3
1.3 Objetivos	6
1.4 Estructura de la tesis	9
2 Antecedentes	11
2.1 Introducción	13
2.2 MOP	13
2.3 MOEAs	17
2.3.1 Marcos conceptuales de los MOEAs	18
2.3.2 Selección y actualización de población	22
2.3.3 Reproducción	23
2.3.4 Características deseables para un MOEA	24
2.4 Soluciones casi-óptimas	27
2.5 Toma de decisiones multicriterio	30

2.6 Métricas . . . . .	31
2.6.1 Métricas de convergencia . . . . .	32
2.6.2 Métrica diversidad . . . . .	33
2.6.3 Métricas convergencia-diversidad . . . . .	34
3 Algoritmo nevMOGA . . . . .	37
3.1 Introducción . . . . .	39
3.2 Nuevo conjunto de interés . . . . .	40
3.3 Discretización del nuevo conjunto de interés . . . . .	42
3.4 Descripción de nevMOGA . . . . .	45
3.5 Evaluación . . . . .	52
3.5.1 Métrica . . . . .	53
3.5.2 Benchmark 1 . . . . .	53
3.5.3 Benchmark 2 . . . . .	59
3.6 Conclusiones . . . . .	62
4 Particularidades del uso de nevMOGA en problemas de optimización multiobjetivo . . . . .	65
4.1 Introducción . . . . .	67
4.2 Multimodalidades . . . . .	67
4.3 Agregación de objetivos de diseño . . . . .	69
4.4 Exclusión de objetivos de diseño . . . . .	72
4.5 Sobreparametrización . . . . .	77
4.6 Conclusiones . . . . .	80
5 Aplicación de nevMOGA a problemas en ingeniería de control . . . . .	83
5.1 Introducción . . . . .	85
5.2 Ejemplo de diseño de un controlador multivariable . . . . .	87
5.3 Diseño del control para la columna de destilación Wood & Berry . . . . .	93
5.4 Diseño del control de un sistema de refrigeración (concurso CIC2018) . . . . .	102
5.4.1 Problema de optimización con un único objetivo . . . . .	104
5.4.2 Problema de Optimización Multiobjetivo . . . . .	111

5.5	Identificación del sistema de refrigeración de un sistema $\mu$ -CHP basado en una pila de PEMFC . . . . .	119
5.5.1	Planteamiento del problema de optimización multiobjetivo . . . . .	120
5.6	Control del sistema de refrigeración de un sistema $\mu$ -CHP basado en una pila de PEMFC . . . . .	131
5.7	Conclusiones . . . . .	145
6	Conclusiones y trabajos futuros . . . . .	149
6.1	Conclusiones . . . . .	151
6.2	Trabajos futuros . . . . .	153
A	Modelo de un sistema de refrigeración para un sistema $\mu$ -CHP . . . . .	157
A.1	Introducción . . . . .	159
A.2	Submodelo 1 (Sub1) . . . . .	163
A.3	Submodelo 2 (Sub2) . . . . .	165
A.4	Submodelo 3 (Sub3) . . . . .	167
A.5	Modelo completo . . . . .	170
	Bibliografía . . . . .	171



## Capítulo 1

# Motivación y objetivo de la tesis



## 1.1 Introducción

En este capítulo se presentan la motivación y los objetivos de la tesis. Esta tesis busca contribuir en el campo de la optimización multiobjetivo aplicado a problemas de identificación y control de sistemas multivariables. Además, en este capítulo también se presenta la organización de la tesis, introduciendo brevemente el contenido de cada uno de los capítulos que la componen.

## 1.2 Motivación

Hoy en día, es común resolver problemas de ingeniería mediante técnicas de optimización [76]. En estos problemas es típico que existan objetivos que entran en conflicto. Un ejemplo de ellos, son los problemas de diseño de sistemas de control [93] [34] y ajuste de parámetros de modelos [67] [66], en el área de ingeniería de sistemas [46]. Estos problemas, especialmente si son multivariables, suelen considerar diversos objetivos a optimizar. Por ello, resulta razonable y natural resolverlos mediante problemas de optimización multiobjetivo (MOP, por sus siglas en inglés, [60] [21] [81]). Mediante este enfoque (normalmente) no se obtiene una única solución, sino un conjunto de alternativas óptimas (frente de Pareto) donde ninguna es mejor al resto en todos los objetivos de diseño. De este modo, el diseñador deberá analizar las prestaciones de las soluciones del frente de Pareto, con el objetivo de tomar la decisión final. Esta metodología proporciona al diseñador un mayor conocimiento del problema, con el propósito de poder elegir, con mayor criterio, la solución más idónea en base a sus preferencias.

Un MOP consta de tres etapas [16] [80]:

1. Definición del problema: En esta etapa se deben definir las variables de decisión, los objetivos de diseño, así como las restricciones, si las hubiera.
2. Proceso de optimización: el MOP debe traducirse en un problema de optimización donde un algoritmo multiobjetivo debe calcular el conjunto de soluciones óptimas.
3. Etapa de decisión multicriterio: En esta etapa, el diseñador se involucra activamente analizando el intercambio entre prestaciones y rendimiento de las diferentes soluciones óptimas encontradas, con el objetivo de elegir la solución más idónea a sus especificaciones.

En un MOP, generalmente el diseñador analiza exclusivamente las soluciones del frente de Pareto. Además de estas soluciones, pueden existir soluciones que pueden ser útiles para el diseñador, como las soluciones multimodales o casi-óptimas, y que suelen ser ignoradas. Dos alternativas multimodales son aquellas soluciones óptimas que tienen los mismos valores en los objetivos de diseño. Es común que un MOP tenga soluciones multimodales cuando algún o algunos objetivos de diseño se componen de objetivos agregados. Cuando esto ocurre, generalmente, debido a que los algoritmos de optimización suelen obtener solo una de ellas (aunque es cierto que existen algoritmos que tienen esta situación presente [20]), el diseñador desconoce esta información. Por ello, el diseñador se ve abocado a tomar la decisión final sin disponer de toda la información relevante, lo que puede implicar una decisión final mejorable. Una posibilidad de obtener estas soluciones multimodales es mediante la desagregación de los objetivos o la agregación de un objetivo adicional [23]. Una solución casi-óptima (también conocida como epsilon-eficiente o aproximada) es aquella que, sin ser óptima, presenta prestaciones similares a las óptimas. Estas soluciones son ignoradas en los MOP clásicos y pueden aportar soluciones con características significativamente diferentes a las óptimas, lo que es una información valiosa para el diseñador antes de llevar a cabo la decisión final.

Las alternativas casi-óptimas pueden resultar interesantes cuando:

1. Existe incertidumbre en los propios objetivos de diseño. En este escenario, un simple cambio en la configuración de ciertos parámetros propios del entorno de simulación (método de integración, tiempo de muestreo, etc) puede provocar que alternativas casi-óptimas puedan dominar a las óptimas. Por ello, en este contexto, las alternativas casi-óptimas son consideradas (prácticamente) igual de interesantes que las óptimas.
2. Cuando queremos estudiar a posteriori nuevas características no contempladas en la fase de optimización. Estas nuevas características se incorporarán en fase de decisión, en forma de nuevos objetivos a estudiar, debido a:
  - Decisión del diseñador: Este es el caso, por ejemplo, de los objetivos relativos a la validación de modelos o controladores. Estos objetivos no son incluidos en los objetivos de diseño porque que no se desea que su información se utilice en la obtención de las alternativas óptimas. Sin embargo, en muchas ocasiones su información puede ser utilizada en la etapa de decisión. De nuevo, es muy útil obtener el conjunto de alternativas casi-óptimas, ya que pueden tener un rendimiento

significativamente mejor en la validación siendo ambas muy similares en los objetivos de diseño.

- Limitación de recursos: Hay objetivos que no se pueden incluir en la fase de optimización debido a limitaciones de recursos, de tiempo o económicos. Un ejemplo de ello, son los objetivos de robustez en diseño de controladores para sistemas no lineales. Estos objetivos suelen ser muy costosos computacionalmente. En este contexto, se puede eliminar este objetivo de la fase de optimización, e incorporarlo a la etapa de decisión, analizándolo solo sobre el conjunto de soluciones obtenidas en el proceso de optimización. De esta forma se reduce considerablemente el coste computacional. Otro ejemplo de esta problemática puede estar en el diseño de controladores donde se desee validar los controladores sobre una planta real. Es común que validar todos los controladores generados durante la fase de optimización no sea posible debido a algunas limitaciones (recursos, tiempo o económicos). Por ello, es factible validar solo los controladores seleccionados para la etapa de decisión. En los dos contextos descritos, las alternativas casi-óptimas pueden proporcionarnos soluciones más robustas, o con mejor rendimiento sobre la planta real, que las óptimas.
  - Ayuda a la decisión: Es posible que determinadas soluciones casi-óptimas obtengan un mejor rendimiento en alguna característica, deseable para el diseñador, no incluida en los objetivos de diseño. Por ejemplo, en el diseño de controladores, un controlador casi-óptimo puede obtener una respuesta con menor tiempo de establecimiento que uno óptimo, si esta característica no ha sido incorporada en los objetivos de diseño. Por ello, el diseñador puede tener una preferencia sobre determinadas soluciones casi-óptimas, en detrimento de las óptimas, lo que implica que los objetivos no han sido elegidos adecuadamente.
3. Cuando los objetivos de diseño están compuestos por la agregación de varios objetivos. Por ejemplo, en los sistemas de control multivariables, los objetivos aumentan considerablemente, y por ello, es común agregar objetivos y simplificar la etapa de optimización y la fase de decisión. Cuando esto ocurre, al definir preferencias a priori y agregar los objetivos, se enmascara información. Es común que existan soluciones con rendimiento similar a las óptimas (casi-óptimas), que sin embargo, tienen características diferentes y pueden presentar diferente *trade-off* en los objetivos agregados. Además de esto, puede existir una agregación

indirecta en los objetivos de diseño. Por ejemplo, cuando un objetivo mide el error a lo largo de un ensayo en el modelado o control de sistemas no lineales. Normalmente el ensayo planteado tiene diferentes cambios de referencia/perturbación o entradas al modelo que llevan al proceso a diferentes zonas de trabajo. Esto hace que controladores o modelos buenos para algunas zonas no lo sean en otras y viceversa. Por lo tanto, se produce una agregación indirecta. De nuevo es útil encontrar soluciones que tengan rendimiento similar a las óptimas pero que tengan diferente *trade-off* en los objetivos agregados.

En todos los escenarios descritos es interesante obtener las alternativas casi-óptimas. Gracias a obtener el conjunto de alternativas casi-óptimas es posible: 1) analizar nuevas características no incluidas en la fase de optimización; 2) estudiar la idoneidad de los objetivos planteados, analizando la posible inclusión de estas nuevas características en la fase de optimización. Estas soluciones son consideradas soluciones prácticamente igual de interesantes que las óptimas, y su consideración en un MOP resulta sumamente valiosa. Sin embargo, obtenerlas todas puede aumentar en exceso el número de alternativas posibles, ralentizando el proceso de optimización y complicando en exceso la etapa de decisión. Por ello, es importante considerar las alternativas casi-óptimas que mayor información relevante aporten al diseñador (potencialmente útiles). La contribución de esta tesis consiste en proponer un nuevo enfoque de optimización multiobjetivo que considera las alternativas casi-óptimas potencialmente útiles, aplicado a la identificación de modelos y el diseño de controladores de sistemas multivariables.

### 1.3 Objetivos

Como se ha comentado en la sección anterior, existe un conjunto de alternativas casi-óptimas, útiles para el diseñador, que son obviadas en un MOP tradicional. El objetivo principal de la tesis es desarrollar una nueva metodología capaz de considerar estas soluciones, aplicada al control y modelado de sistemas multivariables.

Este objetivo principal conlleva a su vez diferentes objetivos parciales:

1. Definición del conjunto de interés (óptimas y casi-óptimas potencialmente útiles).
2. Diseño del algoritmo que caracterice el conjunto de interés.

### 3. Análisis pormenorizado, del conjunto encontrado, en la etapa de decisión.

En relación con el primer objetivo parcial, es necesario obtener un conjunto acotado de soluciones. En muchos MOP existen un gran número de alternativas casi-óptimas. Sin embargo, no todas aportan información nueva al diseñador. Por ello, es necesario obtener solo el conjunto de alternativas casi-óptimas que mayor información relevante aporte (potencialmente útiles), con el fin de no obtener un conjunto con excesivas soluciones posibles. En [89] se dice literalmente:

*“the motivation for the storage of approximate solutions is to obtain in addition to the ‘optimal’ points also points which are close to these points in image space but which differ significantly in parameter space.”*

Es decir, la motivación de almacenar el conjunto de soluciones casi-óptimas (aproximadas) es obtener además de las soluciones óptimas también las soluciones que siendo similares en el espacio de objetivos, difieren significativamente en el espacio de parámetros. Este razonamiento que nosotros compartimos, nos hace creer que las soluciones casi-óptimas potencialmente útiles son las que tienen características diferentes a las que son mejores que ellas, es decir, las alternativas que son significativamente diferentes a las que le dominan.

Veamos un ejemplo concreto. Suponemos que tenemos dos controladores muy similares en sus parámetros, esto es, vecinos. Estos controladores tendrán un comportamiento similar. Si uno de ellos domina ligeramente al otro, el peor de ellos, bajo nuestro criterio, no aporta información relevante al diseñador. El diseñador siempre elegirá el controlador no dominado, ya que teniendo características similares es mejor en los objetivos de diseño. Ahora supongamos que disponemos de dos controladores con parámetros significativamente diferentes, esto es, no vecinos. Si uno de ellos domina ligeramente al otro, ambos controladores son potencialmente útiles. Teniendo ambos un rendimiento similar en los objetivos de diseño, sus diferentes características (debido a tener parámetros significativamente diferentes) aportan nueva información relevante al diseñador. Incluir en los objetivos de diseño, de forma independiente, todas las posibles características deseables de un controlador es muchas veces inabordable. Por ello, es posible que el controlador ligeramente dominado, presente alguna mejora significativa, en cualquier características no incluida (o agregada con otros subjetivos) en los objetivos de diseño. El diseñador deberá analizar estas características para elegir el controlador más adecuado a sus especificaciones.

Con todo ello, bajo nuestro criterio, las alternativas potencialmente útiles son las soluciones óptimas y casi-óptimas no dominadas en su vecindad. Obteniendo solo estas alternativas, obtendremos la información más relevante sin aumentar en exceso el número de ellas.

Una vez definido el conjunto de interés, el siguiente objetivo parcial es diseñar un algoritmo capaz de encontrarlo. Los algoritmos existentes destinados a la localización de alternativas casi-óptimas discretizan únicamente en base al espacio de objetivos. Con este enfoque, una solución casi-óptima puede ser eliminada sin tener en cuenta su ubicación en el espacio de parámetros, y por lo tanto, las alternativas casi-óptimas que tienen características distintas podrían ser descuidadas. Del mismo modo, dichos algoritmos obtendrán un gran número de soluciones casi-óptimas, que bajo nuestro enfoque, no aportan información relevante al diseñador (casi-óptimas dominadas en su vecindad). Por estas razones, los algoritmos existentes no son apropiados para la localización de nuestro conjunto de interés, siendo necesario el diseño de un nuevo algoritmo. Este algoritmo deberá ser capaz de proporcionar el conjunto de alternativas óptimas y casi-óptimas no dominadas en su vecindad.

En este sentido, los algoritmos evolutivos (EAs, por sus siglas en inglés) son herramientas de optimización global y búsqueda de soluciones, que pueden tratar con éxito gran variedad de problemas. En ellos, se mantiene un conjunto de alternativas que son mezcladas y compiten entre sí, de tal manera que evolucionan permitiendo explorar el espacio de búsqueda de una mejor manera. Por estas razones, los EAs multiobjetivo (MOEAs, por sus siglas en inglés) ya han demostrado su potencial en los MOP, ya que encuentran un conjunto de soluciones (frente de Pareto) en una sola ejecución. El algoritmo diseñado en esta tesis se basa en el MOEA ev-MOGA [38]. El objetivo de ev-MOGA es garantizar que el conjunto de soluciones generadas converja hacia el frente de Pareto consiguiendo una caracterización bien distribuida del mismo. El nuevo algoritmo diseñado se evaluará mediante una métrica que mide la convergencia-diversidad del conjunto encontrado sobre dos *benchmarks*.

Una vez se ha comprobado el correcto funcionamiento del algoritmo diseñado, se evaluará su utilidad sobre diferentes MOPs destinados a la identificación de modelos y ajuste de controladores de sistemas multivariables. Como ejemplos de control se utilizarán el diseño de un controlador multivariable, la columna de destilación Wood & Berry y un sistema de refrigeración. Además, se realizará la identificación del modelo y ajuste de controladores del proceso de refrigeración de un sistema de generación de energía térmica y eléctrica basado en una pila de hidrógeno con membrana de intercambio de protones (PEMFC). Una vez el algoritmo ha obtenido el conjunto de controladores o modelos óptimos y casi-

óptimas no dominados en su vecindad, el diseñador, en la etapa de decisión, llevará a cabo un estudio pormenorizado antes de tomar la decisión final. Este enfoque permite el análisis a posteriori de nuevas características no incluidas en la fase de optimización, así como el estudio de nuevas alternativas casi-óptimas con características significativamente diferentes a las óptimas. Con esta nueva información el diseñador puede elegir con mayor criterio, obteniendo información que hubiera sido obviada en un MOP tradicional.

## 1.4 Estructura de la tesis

La organización de la tesis se ha establecido de la siguiente forma:

- **Capítulo 2: Antecedentes.** En este capítulo se presentan los conceptos teóricos relativos a las técnicas de optimización multiobjetivo que servirán para enmarcar los desarrollos planteados en la tesis. En primer lugar, se definen los conceptos básicos de un MOP. Posteriormente, se definen los conceptos necesarios para el estudio de las soluciones casi-óptimas, así como los algoritmos diseñados en la literatura con el propósito de caracterizarlas. Más adelante, se describen algunas herramientas de visualización. Por último, se introducen algunas métricas/indicadores empleados en la literatura para medir el desempeño de los MOEAs.
- **Capítulo 3: nevMOGA.** En este capítulo se presenta un nuevo algoritmo destinado a la obtención de las alternativas óptimas y casi-óptimas no dominadas en su vecindad. Los algoritmos destinados a la localización de alternativas casi-óptimas discretizan solo en función del espacio de objetivos. Esta metodología puede ser adecuada para algunas aplicaciones, sin embargo, una solución casi-óptima puede ser eliminada sin tener en cuenta su ubicación en el espacio de parámetros, y en consecuencia, las soluciones casi-óptimas que tienen características distintas podrían ser descartadas. Esto provocaría una pérdida de soluciones que consideramos especialmente relevantes por la característica de ser diferentes. Por ello, en este capítulo se presenta el nuevo algoritmo diseñado en esta tesis, que tiene como objetivo encontrar las soluciones óptimas y casi-óptimas no dominadas en su vecindad.
- **Capítulo 4: Particularidades del uso de nevMOGA en problemas de optimización multiobjetivo.** Las alternativas obtenidas por nevMOGA pueden ser útiles para el diseñador en cualquier MOP, sin embargo, su utilidad es todavía más relevante ante problemas donde: existen soluciones multimodales, existen objetivos de diseño formados como la agregación de

objetivos parciales, y se excluyen algunos objetivos del proceso de optimización. Además, estas soluciones pueden aportar información relevante que manifieste la existencia de sobreparametrización en el problema definido. En este capítulo se estudian todos estos casos mencionados.

- **Capítulo 5:** Aplicación de nevMOGA a problemas en ingeniería de control. En este capítulo la metodología multiobjetivo introducida previamente se emplea para problemas de identificación y control multivariables. Estos sistemas suelen tener un gran número de objetivos. Al agregar estos objetivos se simplifica la etapa de optimización y la fase de decisión a cambio de perder soluciones óptimas (que pasan a ser casi-óptimas). Gracias a la utilización de nevMOGA es posible la agregación de estos objetivos sin perder apenas alternativas relevantes para el diseñador.
- **Capítulo 6:** Conclusiones. Este capítulo presenta las conclusiones más importantes de la tesis, destacando los resultados obtenidos al aplicar la metodología introducida en esta tesis sobre problemas en el ámbito de la ingeniería de control de procesos. También se presentan las líneas futuras de la tesis.
- **Apéndice A:** En este apéndice se describe el modelo de un proceso de refrigeración para un sistema  $\mu$ -CHP (*Combined Heat and Power*). El modelo ha sido desarrollado a partir de un sistema  $\mu$ -CHP real ubicado en el laboratorio de nuestro equipo de investigación, el grupo de Control Predictivo y Optimización Heurística (CPOH), de la Universitat Politècnica de València. Este modelo será empleado en el capítulo 5 como ejemplo de identificación de un sistema multivariable.

## Capítulo 2

# Antecedentes



## 2.1 Introducción

En este capítulo se presentan los conceptos teóricos relativos con las técnicas de optimización multiobjetivo que servirán para enmarcar los desarrollos planteados en la tesis. En primer lugar, en la sección 2.2, se definen los conceptos básicos de un MOP. En un MOP cabe destacar el proceso de optimización y la etapa de decisión multicriterio. En el proceso de optimización, un algoritmo es necesario para caracterizar adecuadamente el conjunto de alternativas óptimas. En este sentido, los algoritmos evolutivos (EAs [3] [1] [112]) y más concretamente los algoritmos evolutivos multiobjetivo (MOEAs [16] [5] [13]) han demostrado ampliamente su utilidad en la resolución de MOPs. En la sección 2.3 se describirán los MOEAs, se presentará una clasificación de estos algoritmos (en base a su marco conceptual y sus mecanismos de selección, reproducción y actualización) y se describirán sus características deseables. Por otro lado, en la sección 2.4, se introducen los conceptos necesarios para el estudio de las soluciones casi-óptimas, así como los algoritmos diseñados en la literatura con el propósito de caracterizarlas. Respecto a la etapa de decisión multicriterio, en la sección 2.5, se realiza una revisión de las herramientas de visualización más extendidas en la resolución de MOPs. Por último, se introducen algunas métricas/indicadores empleadas para medir las prestaciones de los MOEAs.

## 2.2 MOP

En esta sección se introducen formalmente los conceptos básicos de un MOP: dominancia y conjunto/frente de Pareto. Un problema de optimización multiobjetivo (MOP)<sup>1</sup> puede definirse como:

$$\begin{aligned} \min \quad & \mathbf{f}(\mathbf{x}) \\ \mathbf{x} \in \quad & Q \end{aligned} \tag{2.1}$$

sujeto a:

$$\underline{x}_i \leq x_i \leq \overline{x}_i, \quad i = [1, \dots, k] \tag{2.2}$$

---

<sup>1</sup>Un problema de maximización se puede convertir en una minimización. Para cada uno de los objetivos que hay que maximizar, se puede aplicar:  $\max f_i(\mathbf{x}) = -\min(-f_i(\mathbf{x}))$ .

donde  $\mathbf{x} = [x_1, \dots, x_k]$  es definido como el vector de parámetros en el dominio  $Q \subset \mathbb{R}^k$  y  $\mathbf{f}: Q \rightarrow \mathbb{R}^m$  es definido como el vector de funciones objetivos  $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_m(\mathbf{x})]$ .  $\underline{x}_i$  y  $\overline{x}_i$  son los límites inferior y superior de cada componente de  $\mathbf{x}$ .

Normalmente los objetivos de un MOP entran en conflicto, y por ello, no suele existir una única solución óptima. De esta forma, soluciones que minimizan un objetivo tendrán peor rendimiento en algún otro objetivo y viceversa.

**Definición 2.1** (Dominancia [70]): Un vector solución  $\mathbf{x}^1$  es dominado por otro vector solución  $\mathbf{x}^2$  si  $f_i(\mathbf{x}^2) \leq f_i(\mathbf{x}^1)$  para todo  $i \in [1, \dots, m]$  y  $f_j(\mathbf{x}^2) < f_j(\mathbf{x}^1)$  para al menos un  $j$ ,  $j \in [1, \dots, m]$ . Esto es denotado como  $\mathbf{x}^2 \preceq \mathbf{x}^1$ .

**Definición 2.2** (Conjunto de Pareto): El conjunto de Pareto (denotado como  $P_Q$ ) es el conjunto de soluciones en  $Q$  no dominadas por otra solución en  $Q$ :

$$P_Q := \{\mathbf{x} \in Q \mid \nexists \mathbf{x}' \in Q : \mathbf{x}' \preceq \mathbf{x}\}$$

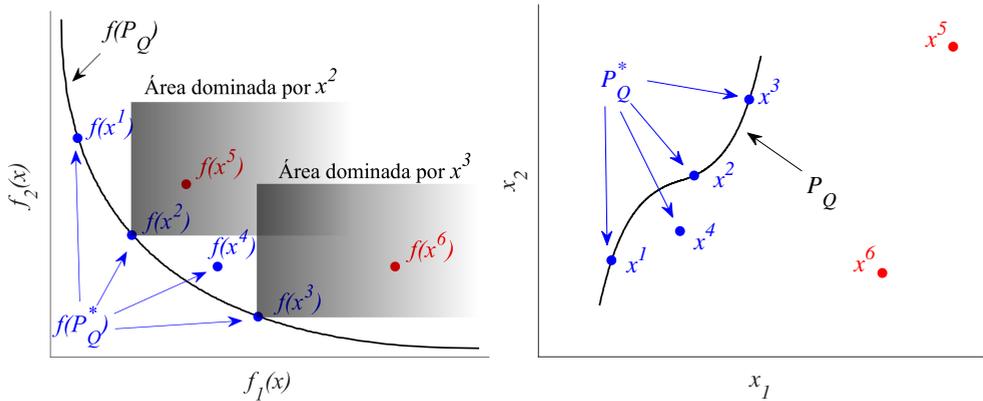
**Definición 2.3** (Frente de Pareto): Dado el conjunto de Pareto  $P_Q$ , el frente de Pareto  $\mathbf{f}(P_Q)$  es definido como:

$$\mathbf{f}(P_Q) := \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in P_Q\}$$

Por lo tanto, se define un MOP como el problema de encontrar el conjunto de soluciones óptimas que minimizan los objetivos de diseño. Estas soluciones han sido definidas formalmente como el frente ( $\mathbf{f}(P_Q)$ ) y conjunto ( $P_Q$ ) de Pareto en su imagen en el espacio de objetivo y parámetros respectivamente.

Generalmente es suficiente con obtener un conjunto discreto  $P_Q^* \subset P_Q$ , de manera que,  $P_Q^*$  caracterice adecuadamente a  $P_Q$ . Esto se debe a que determinar  $P_Q$  es habitualmente inabordable, ya que puede tener infinitas soluciones. Cabe destacar que el conjunto  $P_Q^*$  no es único.

En la figura 2.1 se observa un caso particular de un MOP con dos objetivos  $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x})]$  y dos variables de decisión  $\mathbf{x} = [x_1 \ x_2]$ . En primer lugar, se observa el frente y conjunto de Pareto en color negro. Se tienen 6 posibles soluciones ( $\mathbf{x}^1$  a  $\mathbf{x}^6$ ) al MOP. Cada una de ellas tiene su representación en el espacio objetivo ( $\mathbf{f}(\mathbf{x}^1)$  a  $\mathbf{f}(\mathbf{x}^6)$ ).  $\mathbf{x}^1$  obtiene el mejor rendimiento respecto al objetivo  $f_1$ . Sin embargo, esta solución obtiene el peor rendimiento respecto al segundo objetivo  $f_2$ . Lo contrario ocurre con  $\mathbf{x}^3$ . Por su parte,  $\mathbf{x}^2$  es peor que



**Figura 2.1:** Ejemplo de un MOP con dos objetivos y dos variables de decisión. Concepto de dominancia.

$\mathbf{x}^1$  en el objetivo  $f_1$  pero mejor en  $f_2$ . Lo contrario ocurre respecto a  $\mathbf{x}^3$ . Por lo tanto,  $\mathbf{x}^1$ ,  $\mathbf{x}^2$  y  $\mathbf{x}^3$  forman parte de  $P_Q$ . Por otro lado,  $\mathbf{x}^4$  no forma parte de  $P_Q$ , pero sí del conjunto discreto  $P_Q^*$ , ya que no se ha encontrado ninguna solución que le domine en este caso particular.  $\mathbf{x}^5$  y  $\mathbf{x}^6$  son dos soluciones dominadas (peores en ambos objetivos) por  $\mathbf{x}^2$  y  $\mathbf{x}^3$  respectivamente. Por ello, estas dos alternativas no forman parte del frente de Pareto. Un posible conjunto  $P_Q^*$  podría ser el obtenido en este caso particular, formado por  $\mathbf{x}^1$ ,  $\mathbf{x}^2$ ,  $\mathbf{x}^3$  y  $\mathbf{x}^4$ . Evidentemente no es la única posibilidad, ni seguramente la mejor, de caracterizar  $P_Q$ .

Una vez introducidos los conceptos básicos de dominancia y conjunto/frente de Pareto, se definen las etapas de un MOP. Estas etapas son: la definición del problema, el proceso de optimización y la etapa de decisión multicriterio.

1. En la primera etapa se define el problema: variables de decisión, objetivos de diseño, restricciones y requerimientos del problema. En esta etapa se deben concretar los parámetros (variables de decisión) y el rendimiento particular de cada alternativa de diseño (objetivos). Esta definición no es una tarea sencilla, ya que el diseñador debe adecuar la formulación al punto de vista del optimizador. De la definición del problema depende el conjunto óptimo encontrado, y en consecuencia, la elección final.
2. En la segunda etapa, el problema multiobjetivo debe abordarse como un problema de optimización. En esta etapa debe elegirse el algoritmo multiobjetivo a emplear, que será el encargado de caracterizar el frente

de Pareto. En este sentido, los EAs y más concretamente los MOEAs son una herramienta muy utilizada para esta tarea.

3. Finalmente, en la etapa de decisión multicriterio, el diseñador deberá elegir la solución final. Esta tarea no es trivial. El diseñador debe analizar el intercambio de prestaciones de las diferentes soluciones, con el fin de elegir la solución más idónea a sus especificaciones. Por ello, en muchas ocasiones, sobre todo cuando existen más de 2-3 objetivos, son necesarias herramientas que faciliten la visualización y el análisis del conjunto de alternativas encontradas.

Según como se combinen la resolución de estas etapas, los MOP pueden clasificarse en [85]: a priori, a posteriori o progresivos.

1. En un enfoque a priori, las etapas de optimización y toma de decisiones se llevan a cabo al mismo tiempo, lo que da como resultado una solución única, ya que las preferencias deseadas se definen de antemano. Por ello, este enfoque no permite el análisis del conjunto de alternativas óptimas, lo que proporciona al diseñador menor información. Al obtener una única solución, el proceso debe volver a iniciarse si la solución encontrada no es deseada.
2. En un enfoque a posteriori, la búsqueda no proporciona una solución única, sino un conjunto de soluciones óptimas (frente de Pareto). Este procedimiento requiere más tiempo, pero le proporciona al diseñador información sobre diferentes soluciones y permite tomar una decisión con mayor criterio.
3. Por último, en el enfoque progresivo la búsqueda y toma de decisiones se realiza de forma interactiva. El algoritmo de optimización involucra al diseñador en el proceso de optimización ajustando sus preferencias interactivamente. Esta podría ser una característica deseable para un algoritmo cuando el diseñador tiene conocimiento del intercambio entre los objetivos de diseño. En este enfoque, las preferencias son utilizadas en el proceso de optimización, lo que requiere mayor tiempo para el diseñador.

## 2.3 MOEAs

En el proceso de optimización de un MOP es necesario el uso de un algoritmo para caracterizar el conjunto de soluciones óptimas. En este sentido, una alternativa muy interesante son los MOEAs. Se trata de técnicas de optimización estocásticas basadas en la teoría evolutiva NeoDarwiniana [3]. Los MOEAs utilizan una población de individuos que se hace evolucionar en un proceso iterativo y generacional. Estos algoritmos realizan una búsqueda paralela que abre la opción de encontrar un conjunto de alternativas en una sola ejecución. Los algoritmos genéticos multiobjetivo (MOGA [62], por sus siglas en inglés) son una de las técnicas más populares entre los MOEAs. Estos algoritmos fueron creados para resolver multitud de problemas, lo que ha potenciado su uso [83] [110] [111] [33].

Los MOEAs evolucionan constantemente. Su gran ventaja reside en que es una técnica robusta y muy flexible. Pese a ello, no garantizan la localización del frente de Pareto. Sin embargo, existen evidencias de que consiguen soluciones aceptables en un tiempo razonable, comparado con otro tipo de algoritmos [13]. Los MOGAs crean nuevos individuos mediante reglas de selección, operadores de recombinación (cruce) y mutación. De esta forma, las soluciones se reproducen en cada iteración, manteniendo las mejores de ellas, evolucionando hacia una población que converja hacia el frente de Pareto.

Las características principales de los MOGAs son [37]:

1. Amplia aplicabilidad de problemas a resolver.
2. Simplicidad conceptual.
3. Pueden paralelizarse fácilmente (puede reducir significativamente el coste computacional).
4. Poco sensibles a mínimos locales.
5. Pueden caracterizar frentes de Pareto discontinuos.

Un ejemplo de estructura básica de un MOEA se observa en el algoritmo 1. Un MOEA consta de una Población  $P(t)$  inicializada con  $N$  individuos. Se selecciona una población  $P'(t)$  perteneciente a  $P(t - 1)$ .  $P'(t)$  es evolucionado mediante la alteración de sus genes utilizando para ello operadores de cruce y mutación. La operación de cruce realiza un intercambio de información genética entre los individuos, mientras que la mutación introduce cambios en algunos individuos. Una vez evaluado  $P'(t)$  se analiza la agregación o no de sus

**Algoritmo 1** Algoritmo Evolutivo Multiobjetivo Simple

---

```
1:  $t := 0$ ;  
2: Inicializar Población  $P(t)$   
3: Evaluar Población  $P(t)$   
4: while  $t < t_{max}$  do  
5:    $t := t + 1$   
6:   Seleccionar  $P'(t)$  de  $P(t - 1)$   
7:   Alterar  $P'(t)$   
8:   Evaluar Población  $P'(t)$   
9:   Determinar  $P(t)$  a partir de  $P(t - 1)$  y  $P'(t)$   
10: end while
```

---

individuos en la población  $P(t)$ . Este proceso es iterativo y finalizará cuando se alcancen el número de generaciones definidas por el usuario.

Los MOEAs son muy utilizados en los problemas de optimización multiobjetivo debido a sus características y a la posibilidad de obtener el frente de Pareto en una sola ejecución. En los MOPs encontrar el frente de Pareto no es una tarea trivial. Este proceso tiene un coste computacional elevado. Por ello, muchos autores han desarrollado MOEAs destinados a multitud de aplicaciones, para converger más eficientemente hacia un frente de Pareto adecuadamente distribuido.

### 2.3.1 Marcos conceptuales de los MOEAs

La mayoría de MOEAs utilizan de forma iterativa: 1) un operador de selección basado en la dominancia de Pareto y en la densidad de soluciones, 2) un operador de reproducción y 3) un proceso de actualización basado en la dominancia de Pareto y en la densidad de soluciones. Los MOEAs más eficientes utilizan el concepto de elitismo [22]. Los MOEAs elitistas utilizan la aproximación al frente de Pareto, obtenida en cada iteración, activamente en el proceso de optimización. De esta forma se facilita la exploración, utilizando la información ya obtenida. Los algoritmos elitistas garantizan que los individuos mejores (los no dominados) no pueden ser eliminados de la población en favor de individuos peores (dominados). Por lo tanto, los MOEAs elitistas consiguen: 1) converger más eficientemente al frente de Pareto y 2) evitar la pérdida de soluciones no dominadas por alternativas peores.

Algunos de los MOEAs elitistas más empleados son: NSGA-II [22], SPEA2 [119] o PESA-II [17]. NSGA-II, una versión mejorada de NSGA, hace competir a las soluciones existentes con su descendencia (individuos generados a partir de ellas). Este algoritmo tiene una baja complejidad y consigue una

adecuada diversidad de soluciones sin necesidad de parámetros adicionales. El algoritmo SPEA2 es una versión mejorada de SPEA. Este algoritmo utiliza dos poblaciones: una población principal y una población o archivo donde se almacena las soluciones no dominadas. SPEA2 utiliza técnicas de *clustering* [72] para actualizar el archivo que contiene las soluciones óptimas, con el objetivo de mejorar la distribución de las soluciones obtenidas. El algoritmo PESA-II, una versión mejorada de PESA, distribuye el frente de Pareto en *hyperboxes*. PESA-II asigna a cada *hyperbox* un valor según su nivel de dominancia (en lugar de a cada individuo). Este método obtiene una adecuada distribución del frente de Pareto, reduciendo el coste computacional en la clasificación de las soluciones.

Existen diferentes marcos conceptuales que los autores han definido para el diseño y clasificación de los MOEAs [117]:

1. Basados en descomposición.
2. Basados en preferencias.
3. Basados en Indicadores.
4. Híbridos.
5. Meméticos.
6. Basados en coevolución.

#### *MOEAs basados en descomposición*

Este tipo de algoritmos son relativamente recientes. El MOP se descompone en subproblemas con un único objetivo. Cada subproblema tiene una agregación ponderada, lineal o no lineal, de los objetivos de diseño, y por tanto, tiene una única solución. El algoritmo genera nuevas soluciones para cada subproblema mediante operadores genéticos, y con ellas, se actualiza la solución al subproblema. Los subproblemas utilizan información de los subproblemas vecinos, es decir, de los subproblemas con vector de pesos similar. El algoritmo elitista MOEA/D [115] es un ejemplo de este enfoque. MOEA/D ha sido empleado con éxito para muchas aplicaciones [12] [68] [58] y diferentes autores han propuesto algunas modificaciones [50] [116] [69] a este algoritmo.

### *MOEAs basados en preferencias*

Un MOP puede tener un número muy elevado (o incluso infinito) de soluciones debido al conflicto entre objetivos. Sin embargo, en muchas ocasiones el diseñador está interesado en una zona preferida. Los métodos de resolución de los MOP que tienen en cuenta las preferencias del diseñador antes de la etapa de decisión son: el enfoque a priori y el enfoque progresivo. En un enfoque a priori las preferencias del diseñador son definidas antes del proceso de optimización. En un enfoque progresivo, algunos resultados intermedios se presentan al diseñador para comprender mejor el problema. Posteriormente, el diseñador proporciona información sobre sus preferencias para guiar la búsqueda hacia la zona preferida. Uno de los primeros algoritmos que incluía las preferencias del diseñador se presentó en [32]. En este algoritmo se define el rango de los miembros de la población mediante la dominancia de Pareto y la información de preferencia del diseñador. En [85] se propone un MOEA que integra las preferencias a priori del diseñador. Las preferencias se expresan en niveles de interés (deseable, tolerable, indeseable,...). Con ello, se define un problema con un único objetivo de diseño sin necesidad de definir los pesos de la agregación.

### *MOEAs basados en indicadores*

La calidad de un frente de Pareto discreto puede medirse mediante un indicador, como la distancia generacional o el hipervolumen (ver sección 2.6). Estos MOEAs utilizan un indicador para la selección de soluciones, y de esta forma, guiar la búsqueda. Uno de los primeros MOEA que utiliza este enfoque es el *indicator-based evolutionary algorithm* (IBEA [118]). Este algoritmo elitista utiliza un indicador arbitrario para comparar pares de soluciones candidatas. IBEA no requiere de otros mecanismos de preservación de la diversidad. Un algoritmo evolutivo basado en el hipervolumen se presenta en [4] (HypE). El cálculo del hipervolumen tiene un coste computacional que crece exponencialmente con el número de objetivos, y por ello, es inabordable con un gran número de ellos. Este algoritmo reduce considerablemente el coste computacional utilizando simulaciones de Monte Carlo para estimar el valor del hipervolumen de un conjunto. De esta forma, HypE puede aplicarse a MOPs con muchos objetivos.

### *Híbridos*

Para el diseño de MOEAs se emplean muchas técnicas diferentes. Cada una de ellas tiene diferentes ventajas y está destinada a la resolución de un tipo de problemas. Por ello, parece razonable hibridar diferentes técnicas, con el objetivo de combinar las ventajas de cada una de ellas, ampliando sus beneficios y posibilidades de aplicación a un mayor número de problemas. La hibridación puede producirse en los métodos de búsqueda y/o los métodos de actualización. La hibridación en los métodos de búsqueda combina métodos de búsqueda global y local [49]. También es posible combinar diferentes operadores de búsqueda. El proceso de búsqueda puede dividirse en diferentes fases, utilizando diferentes estrategias en cada una de ellas [109]. La hibridación en la actualización combina diferentes estrategias de diferentes algoritmos y generalmente se combina con la hibridación de los métodos de búsqueda [28].

### *Meméticos*

Esta es una estrategia reciente, incorpora la búsqueda local y es capaz de ofrecer mejor velocidad y precisión en las soluciones finales [44] [74]. En los algoritmos meméticos, la búsqueda global viene dada por la naturaleza evolutiva del enfoque, mientras que la búsqueda local se utiliza para refinar soluciones a problemas difíciles. Se han aplicado con éxito a muchos problemas [47] [43] [113]. En [42] se propone uno de los primeros algoritmos con este enfoque. Este algoritmo utiliza la búsqueda local una vez aplicados los operadores de variación clásicos. Esta búsqueda local se realiza mediante una suma ponderada aleatoria de los objetivos, es decir, diferente en cada búsqueda de soluciones. MOEA/D [115] puede considerarse uno de estos algoritmos, ya que separa el problema en diferentes subproblemas de un único objetivo.

### *Basados en coevolución*

La coevolución en un MOEA puede considerarse como la evolución de múltiples subpoblaciones simultáneamente para resolver el MOP. De esta forma, se define un sistema cooperativo y competitivo de ayuda, produciéndose descomposiciones del problema y aprovechando correlaciones y dependencias. Los algoritmos basados en coevolución evolucionan una o varias poblaciones auxiliares además del archivo (frente de Pareto discreto) al mismo tiempo. Con este enfoque se puede conseguir una convergencia más rápida al frente de Pareto. Un ejemplo de algoritmo basado en coevolución es ev-MOGA (ver algoritmo 2 [38]). Este algoritmo es un MOEA elitista. El objetivo de ev-MOGA es garan-

tizar que el conjunto de soluciones generadas converja hacia el conjunto óptimo de Pareto utilizando memoria limitada, y al mismo tiempo, consiguiendo una buena distribución del conjunto objetivo a lo largo del frente de Pareto. Este algoritmo consta de tres poblaciones:  $P(t)$ ,  $A(t)$  y  $G(t)$ .  $P(t)$  es la población principal y se encarga de explorar el espacio de búsqueda.  $A(t)$  es el archivo donde se almacenan la aproximación al frente de Pareto.  $G(t)$  es la población auxiliar que se utiliza para almacenar los nuevos individuos generados en cada iteración. De esta forma, la población  $P(t)$  es inicializada aleatoriamente. Una vez generada, se crea el archivo  $A(t)$  (aproximación al frente de Pareto). Posteriormente, en cada iteración, se crea la población  $G(t)$  (reproduciendo individuos de  $P(t)$ ), se evalúa su incorporación al archivo  $A(t)$  y se actualiza el archivo  $P(t)$  con los nuevos individuos de  $G(t)$ .

---

### Algoritmo 2 Algoritmo ev-MOGA

---

```

1:  $t := 0$ 
2:  $A(t) := \emptyset$ 
3:  $P(t) := \text{inicializar}(D)$  ▷ aleatoriamente
4:  $\text{evaluar}(P(t))$ 
5:  $A(t) := \text{actualizar}(P(t), A(t))$ 
6: while  $t < t\_max$  do
7:    $G(t) := \text{crear}(P(t), A(t))$ 
8:    $\text{evaluar}(G(t))$ 
9:    $A(t+1) := \text{actualizar}(G(t), A(t))$ 
10:   $P(t+1) := \text{actualizar}(G(t), P(t))$ 
11:   $t := t + 1$ 
12: end while

```

---

#### 2.3.2 Selección y actualización de población

La selección de individuos a evolucionar y la actualización de la población (o poblaciones) en los MOEAs es un proceso importante. En un problema con un único objetivo este proceso puede ser sencillo. Sin embargo, en un MOP este proceso puede volverse más complejo al disponer de un conjunto de soluciones óptimas. Generalmente en los MOEA existen dos etapas para la selección de individuos y la actualización de la población [117]. En la primera etapa se clasifican las soluciones en base a su nivel de dominancia (*domination rank* [97], *domination count* [32], *domination strength* [120], etc.). De esta forma, las soluciones mejores (no dominadas) tendrán mayor probabilidad de ser elegidas para su evolución y menor probabilidad de ser eliminadas en la actualización. En la segunda etapa se discrimina entre soluciones del mismo nivel de dominancia. Esta discriminación, generalmente, se realiza en base a la densidad de los individuos, es decir, en como de poblado está su entorno en el

espacio de objetivos. Con el objetivo de medir la densidad existen diferentes indicadores (*niching and fitness sharing strategy* [32], *crowding distance* [22], *fast sorting* [75], ...). Se prefieren soluciones menos densas, es decir, con pocos individuos en su entorno próximo (espacio de objetivos), para contribuir a la diversidad de las mismas.

### 2.3.3 Reproducción

Una vez elegidos los individuos a evolucionar, comienza el proceso de reproducción de los mismos para generar nuevas soluciones. Los operadores de reproducción convencionales para EA destinados a problemas de un solo objetivo se pueden extrapolar a MOPs. Sin embargo, la optimización multiobjetivo difiere del enfoque con un solo objetivo, y en consecuencia, es posible que estos operadores no sean adecuados. Diferentes enfoques han sido propuestos para la reproducción de individuos en MOEAs. Los algoritmos de evolución diferencial (DE [99] [79]) utilizan una diferencia ponderada entre soluciones para alterar la población y generar nuevos individuos. Este enfoque fue diseñado para problemas de un único objetivo, pero debido a su simplicidad y eficiencia ha sido utilizado para MOP. Los algoritmos basados en inmunidad [15] utilizan la selección clonal para seleccionar individuos prometedores que son cruzados y mutados para generar nuevos individuos. *Particle swarm optimization* (PSO [27]) es una técnica inspirada en el comportamiento de las aves o los insectos. Los algoritmos que emplean esta técnica, proponen una solución al problema y la evolucionan interactuando con otros individuos. En este proceso, se utiliza información de las soluciones próximas. El enfoque probabilístico basado en modelos extrae información estadística global de la búsqueda anterior y construye un modelo de distribución de probabilidad de individuos prometedores. Este enfoque no utiliza los operadores de cruce o mutación. Un ejemplo de este enfoque es la *Ant Colony Optimization* (ACO [26]). La técnica de *simulated annealing* (SA [84] [96]) es una técnica de optimización global que produce un conjunto de ecuaciones que reproducen la dinámica de un proceso muestreado, mejorando en algunos casos, a otras técnicas de reproducción modernas cuando la evaluación de la función objetivo es limitada.

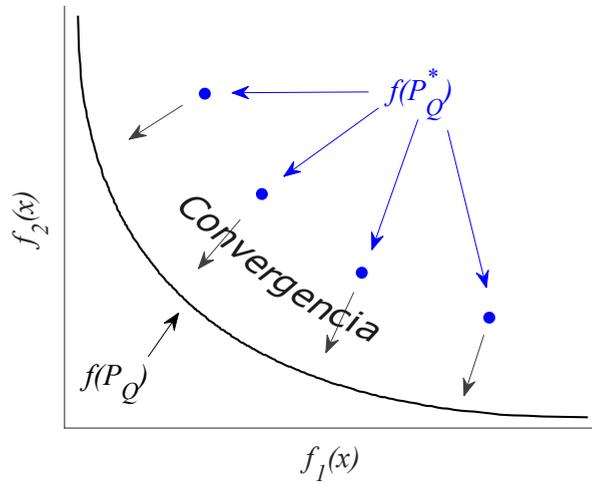
### 2.3.4 Características deseables para un MOEA

Los MOEAS tienen algunas características que podrían ser demandadas por el diseñador o por el MOP en cuestión. Las características deseables para un MOEA están relacionadas con el conjunto de soluciones requeridas por el diseñador y con la definición del problema [78]. Por un lado, obtener un conjunto de alternativas que caracterice correctamente al frente de Pareto permite al diseñador comparar las prestaciones de cada una de ellas y elegir la más idónea con mayor criterio. Es por ello, que los MOEAs que obtienen un adecuado conjunto de alternativas son muy deseables por el diseñador. Por otro lado, existen un conjunto de características para los MOEAs relacionadas con la definición del problema. Estas características serán importantes ante MOPs con diferentes particularidades (con restricciones, multimodales, con muchos objetivos, computacionalmente costosos,...).

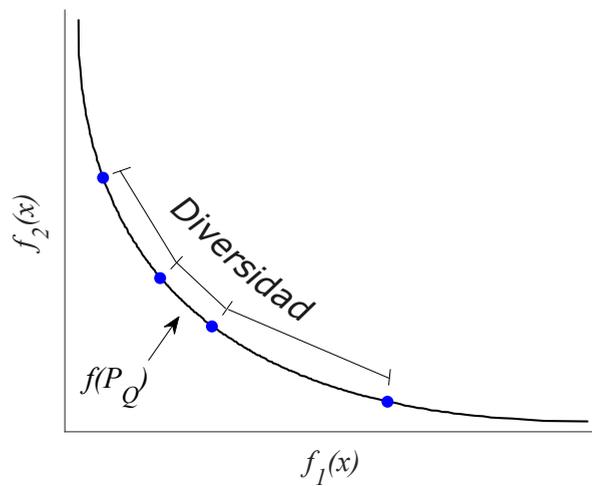
#### *Características relacionadas con el conjunto de soluciones requeridas*

El conjunto de soluciones obtenido por un MOEA debe caracterizar al frente de Pareto. La obtención de un conjunto óptimo adecuado es crucial para el diseñador, ya que su decisión final depende en gran medida de ello. El conjunto de alternativas obtenidas debe tener: convergencia, diversidad y pertinencia.

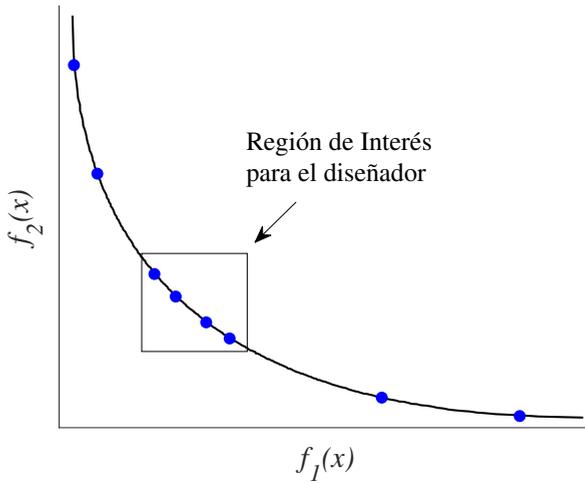
1. La convergencia es la capacidad del algoritmo para alcanzar el frente de Pareto  $P_Q$  (ver figura 2.2). Es decir, es el grado de proximidad en la distancia entre un conjunto discreto de soluciones obtenido  $P_Q^*$  y el frente de Pareto verdadero  $P_Q$ , en el espacio de objetivos. Las propiedades de convergencia usualmente dependen de los parámetros evolutivos del MOEA utilizado. La convergencia es una de las características más importantes para el diseñador en un algoritmo de optimización. Sin embargo, esta característica puede ser insuficiente sin diversidad en el conjunto obtenido.
2. La diversidad es la capacidad del algoritmo de obtener un conjunto de soluciones bien distribuidas en el espacio objetivo. Un conjunto de soluciones adecuadamente distribuidas permite al diseñador analizar el intercambio entre prestaciones y rendimiento de las diferentes soluciones (ver figura 2.3). Sin embargo, una buena diversidad en  $P_Q^*$  no debe confundirse con la pertinencia del conjunto.
3. La pertinencia es la capacidad de obtener un conjunto de soluciones interesantes desde el punto de vista del diseñador (ver figura 2.4). Diferentes técnicas se basan en conseguir un conjunto distribuido uniformemente. Sin



**Figura 2.2:** Convergencia hacia el frente de Pareto.



**Figura 2.3:** Diversidad en el frente de Pareto.



**Figura 2.4:** Pertinencia sobre el frente de Pareto.

embargo, un gran conjunto de soluciones puede no ser de interés para el diseñador [18]. Por lo tanto, para lograr una mayor pertinencia, son deseables algunos mecanismos para incorporar estas preferencias del diseñador [61] [77].

### *Características relacionadas con la definición del problema*

Cualquier MOP puede tener algunas particularidades: restricciones, muchos objetivos, dinámicas, multimodales, robustos, computacionalmente costosos y con muchas variables de decisión. Ante un MOP con cualquiera de estas características es necesario que el MOEA utilizado se adapte bien a este tipo de situaciones.

1. Una gran cantidad de problemas de optimización consideran restricciones. Por ello, estos mecanismos son siempre un tema interesante, desarrollado con diferentes técnicas para la optimización [31] [14].
2. Los algoritmos con buenos mecanismos para la diversidad de soluciones podrían no ser muy eficientes con un gran número de objetivos. Esto se debe a que si las soluciones son resistentes a la dominancia (con un gran número de objetivos), el algoritmo pierde tiempo y recursos en áreas no óptimas [73].

3. En algunas ocasiones, un enfoque estático no es suficiente y el diseñador desea una optimización dinámica [30] [19] cuando la función de coste varía con el tiempo. En este caso, el desafío es seleccionar la solución óptima deseada en cada instante de tiempo.
4. Las soluciones multimodales pueden existir en un MOP. Dos soluciones son multimodales cuando tienen los mismos valores en los objetivos de diseño. Cuando esto ocurre, es deseable que el diseñador obtenga ambas soluciones. Con esta información se podrá seleccionar la más adecuada en cada caso [56].
5. En un MOP, en ocasiones, es también importante el análisis de sensibilidad de las soluciones obtenidas [8]. Dicho con otras palabras, la posible degradación ante la presencia de incertidumbres. Diferentes técnicas se han propuesto con este objetivo.
6. La capacidad de un MOEA para tratar con MOP con un gran número de variables de decisión también es una característica deseable. Un MOEA puede tener un buen funcionamiento con un número limitado de variables de decisión y que su coste computacional aumente considerablemente con un número elevado de parámetros. Por ello es importante que un MOEA tenga un coste computacional razonable en este tipo de problemas [54].

## 2.4 Soluciones casi-óptimas

Extender el conjunto de alternativas obtenidas a aquellas casi-óptimas (también llamadas epsilon-eficientes o aproximadas) aumenta significativamente el número de opciones para el diseñador [89]. El conocimiento de estas alternativas puede proporcionar nuevas soluciones significativamente diferentes a las óptimas, lo que es una información valiosa para el diseñador. Además, también pueden proporcionar una medida de robustez [92]. Diferentes autores hablan de la utilidad de considerar estas alternativas en los MOPs [53] [107]. Varios de ellos utilizan estas alternativas como medio para llegar de una forma más eficiente al frente de Pareto [120] [119], pero pocos son los que proporcionan estas alternativas como posibles soluciones para el diseñador [88] [89] [104].

Para poder estudiar estas soluciones es necesario introducir el concepto de  $-\epsilon$ -dominancia, y el conjunto de alternativas óptimas y casi-óptimas ( $P_{Q,\epsilon}$ ) en su imagen en el espacio de parámetros y objetivos. Una solución  $-\epsilon$ -dominada es aquella que es no dominada considerando la máxima degradación aceptable por el diseñador.

**Definición 2.4** (*− $\epsilon$ -dominancia [88]*): Definido  $\epsilon = [\epsilon_1, \dots, \epsilon_m]$  como la máxima degradación de rendimiento aceptable. Un vector solución  $\mathbf{x}^1$  es  $-\epsilon$ -dominado por otro vector solución  $\mathbf{x}^2$  si  $f_i(\mathbf{x}^2) + \epsilon_i \leq f_i(\mathbf{x}^1)$  para todo  $i \in [1, \dots, m]$  y  $f_j(\mathbf{x}^2) + \epsilon_j < f_j(\mathbf{x}^1)$  para al menos un  $j$ ,  $j \in [1, \dots, m]$ . Esto es denotado como  $\mathbf{x}^2 \preceq_{-\epsilon} \mathbf{x}^1$ .

**Definición 2.5** (*Conjunto  $P_{Q,\epsilon}$  [89]*):  $P_{Q,\epsilon}$  es el conjunto de soluciones en  $Q$  que son no  $-\epsilon$ -dominadas por cualquier otro individuo en  $Q$ :

$$P_{Q,\epsilon} := \{\mathbf{x} \in Q \mid \nexists \mathbf{x}' \in Q : \mathbf{x}' \preceq_{-\epsilon} \mathbf{x}\}$$

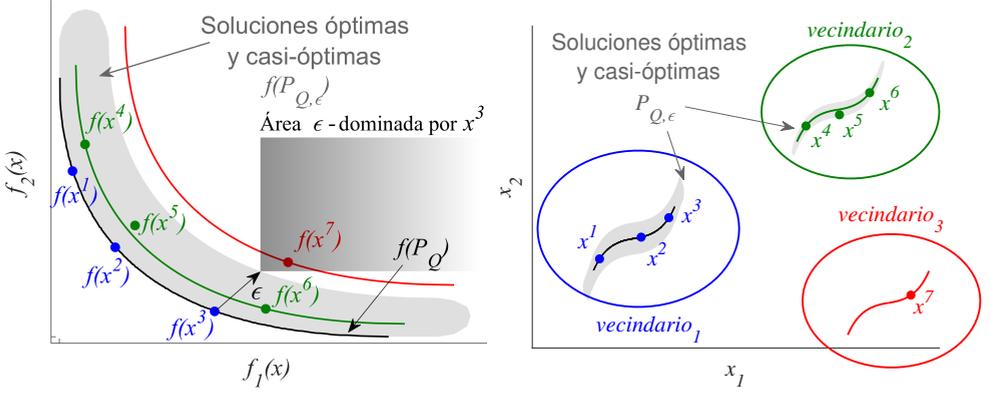
**Definición 2.6** ( *$\mathbf{f}(P_{Q,\epsilon})$* ): Dado un conjunto de soluciones óptimas y casi-óptimas  $P_{Q,\epsilon}$ , se define  $\mathbf{f}(P_{Q,\epsilon})$  como su representación en el espacio objetivo:

$$\mathbf{f}(P_{Q,\epsilon}) := \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in P_{Q,\epsilon}\}$$

Toda solución no  $-\epsilon$ -dominada formará parte del conjunto de alternativas óptimas y casi-óptimas ( $P_{Q,\epsilon}$ ) (ver definición 2.5). De nuevo, habitualmente es suficiente determinar un conjunto discreto  $P_{Q,\epsilon}^* \subset P_{Q,\epsilon}$ , de manera que,  $P_{Q,\epsilon}^*$  caracterice adecuadamente a  $P_{Q,\epsilon}$ . Esto se debe a que determinar  $P_{Q,\epsilon}$  es habitualmente inabordable al tener un conjunto excesivo de soluciones. Cabe destacar que el conjunto  $P_{Q,\epsilon}^*$  no es único.

En la figura 2.5 se observa un caso particular de un MOP con dos objetivos  $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x})]$ , dos variables de decisión  $\mathbf{x} = [x_1 \ x_2]$  y una máxima degradación de rendimiento aceptable  $\epsilon$ . En primer lugar, se observa el frente y conjunto de Pareto en color negro. Se tienen siete posibles soluciones ( $\mathbf{x}^1$  a  $\mathbf{x}^7$ ) al MOP. Cada una de ellas tiene su representación en el espacio objetivo ( $\mathbf{f}(\mathbf{x}^1)$  a  $\mathbf{f}(\mathbf{x}^7)$ ).  $\mathbf{x}^1$ ,  $\mathbf{x}^2$  y  $\mathbf{x}^3$  forman parte del frente de Pareto ( $P_Q$ ).  $\mathbf{x}^4$ ,  $\mathbf{x}^5$  y  $\mathbf{x}^6$  son alternativas dominadas, pero no  $-\epsilon$ -dominadas, y en consecuencia, son soluciones casi-óptimas ( $P_{Q,\epsilon}$ ). En cambio,  $\mathbf{x}^7$  es una solución  $-\epsilon$ -dominada por  $\mathbf{x}^3$ , y por ello, no forma parte de  $P_{Q,\epsilon}$ . En este ejemplo concreto se tiene un conjunto  $P_{Q,\epsilon}^*$  formado por las soluciones  $\mathbf{x}^1$  a  $\mathbf{x}^6$ . Como se ha mencionado anteriormente, encontrar soluciones significativamente diferentes aporta al diseñador nueva información para tomar la decisión final con mayor criterio. Por este motivo, en este ejemplo, cabe destacar la importancia de obtener soluciones del *vecindario*<sub>2</sub> (además, lógicamente, de soluciones del *vecindario*<sub>1</sub>), ya que aportan soluciones potencialmente útiles para el diseñador.

Uno de los primeros algoritmos multiobjetivo destinado a encontrar las soluciones casi-óptimas fue  $P_{Q,\epsilon}$ -NSGA-II [91] (ver algoritmo 3). Este algoritmo se



**Figura 2.5:** Ejemplo de un MOP con dos objetivos y dos variables de decisión. Obtención del Conjunto  $P_{Q,\epsilon}$ .

basa en el MOEA NSGA-II. Su objetivo es encontrar diversidad de alternativas en la zona de soluciones óptimas y casi-óptimas. Inicialmente se crea el archivo  $\mathcal{A}_l$  aleatoriamente (archivo que contiene el conjunto  $P_{Q,\epsilon}^*$ ). Posteriormente, mediante la función  $Select()$  se selecciona un conjunto aleatorio de individuos (de tamaño  $n_p$ ) del archivo  $\mathcal{A}_l$  y se almacenan en  $\mathcal{C}$ . De este conjunto se elegirá el individuo a evolucionar. De esta forma, se fomenta la diversidad de soluciones dentro de toda la zona de soluciones casi-óptimas (no solo del frente de Pareto). Los siguientes tres operadores son idénticos al algoritmo NSGA-II. La función  $DominationSort()$  asigna a  $\mathcal{C}$  un ranking según el nivel de dominancia y la distancia de hacinamiento en el espacio de objetivos (densidad de cada alternativa). Este ranking es almacenado en  $\mathcal{C}'$ .  $TournamentSelection()$  realiza la selección, escogiendo la mejor solución de  $\mathcal{C}'$  ( $\mathcal{P}$ ).  $GeneticOperator()$  se encarga de evolucionar la alternativa  $\mathcal{P}$  generando  $\mathcal{O}$ . Esta evolución se realiza mediante los métodos de cruce y mutación. Finalmente se actualiza el archivo  $\mathcal{A}_l$  por medio de los descendiente calculados  $\mathcal{O}$ .

---

### Algoritmo 3 Iteración de $P_{Q,\epsilon}$ -NSGA-II

---

- 1:  $\mathcal{C} := Select(\mathcal{A}_l, n_p/2)$
  - 2:  $\mathcal{C}' := DominationSort(\mathcal{C})$
  - 3:  $\mathcal{P} := TournamentSelection(\mathcal{C}')$
  - 4:  $\mathcal{O} := GeneticOperator(\mathcal{P})$
  - 5:  $\mathcal{A}_{l+1} := ArchiveUpdateP_{Q,\epsilon}(\mathcal{A}_l, \mathcal{O}, \Delta)$
- 

Sin embargo, este algoritmo no consigue su objetivo: diversidad en el conjunto  $P_{Q,\epsilon}^*$ . La probabilidad de elegir cualquier solución del archivo  $\mathcal{A}_l$  en  $\mathcal{C}$  es la misma. Sin embargo, al clasificar el conjunto  $\mathcal{C}$  según su nivel de dominancia ( $\mathcal{C}'$ )

y elegir el mejor individuo ( $\mathcal{P}$ ) para calcular su descendencia ( $\mathcal{O}$ ), se incorpora la preferencia de evolucionar soluciones de  $P_Q$  (o lo más cercanas a él). Esto impide que las soluciones casi-óptimas tengan la misma probabilidad de tener descendencia que las alternativas del frente de Pareto. Por ello, este algoritmo no consigue una buena diversidad de soluciones en los vecindarios con soluciones casi-óptimas. Por lo tanto, el conjunto  $P_{Q,\epsilon}^*$  obtenido está sesgado por  $P_Q^*$ .

Para resolver este inconveniente se diseñó un nuevo algoritmo  $P_{Q,\epsilon}$ -MOEA [92] (ver algoritmo 4). Su objetivo es encontrar el conjunto de alternativas óptimas y casi-óptimas, evitando el sesgo del frente de Pareto. La estrategia de este algoritmo es muy parecida al  $\epsilon$ -MOEA, basado en archivos de estado estacionario. Se alimenta al archivo inicial con elementos al azar. Posteriormente, se seleccionan al azar dos soluciones de  $P_{Q,\epsilon}^*$ . Estas dos soluciones evolucionan calculando sus descendientes. Finalmente, se actualiza el archivo  $\mathcal{A}_l$ , que contiene el conjunto  $P_{Q,\epsilon}^*$ , con los descendientes calculados. De esta forma, se evita el sesgo de  $P_Q$ . Todas las soluciones del conjunto  $P_{Q,\epsilon}^*$  tienen la misma probabilidad de ser elegidas y con ello se consigue converger hacia  $P_{Q,\epsilon}$  de un modo más eficiente y distribuido.

---

**Algoritmo 4** Iteración de  $P_{Q,\epsilon}$ -MOEA

---

```

1:  $P_0 \in Q$  aleatoriamente
2:  $A_0 := ArchiveUpdateP_{Q,\epsilon}(P_0, 0)$ 
3: for  $l=0,1,2,\dots$  do
4:   Se eligen  $p_1, p_2 \in A_l$  aleatoriamente
5:   Alterar  $p_1, p_2$  para generar  $o_1, o_2$ 
6:    $A_{l+1} := ArchiveUpdateP_{Q,\epsilon}(o_1, o_2, A_l)$ 
7: end for

```

---

## 2.5 Toma de decisiones multicriterio

En la resolución de MOPs se obtiene un frente de Pareto, es decir, un conjunto de soluciones óptimas, donde el diseñador debe elegir la más adecuada a sus preferencias. Con este propósito, generalmente, el diseñador utiliza una herramienta gráfica que permita comparar las prestaciones de cada una de las alternativas encontradas [9]. Las herramientas de visualización tradicionales se vuelven inapropiadas cuando se tiene problemas de más de 2-3 dimensiones. En este contexto, las herramientas de visualización multidimensionales toman un papel clave para ayudar al diseñador a la decisión final.

Las herramientas de visualización multidimensional más comunes son [101]:

- Scatter diagrams [11]: Esta técnica se basa en una serie de diagramas de dispersión situados en forma de una matriz cuadrada, donde cada dimensión define una fila y columna de la matriz. La complejidad de esta representación aumenta significativamente con el número de dimensiones.
- Parallel coordinates [41]: Esta técnica realiza la representación en un gráfico bidimensional. Cada dimensión se traduce en una coordenada del eje de abscisas. Esta visualización es compacta, pero su claridad de representación se pierde con grandes conjuntos de alternativas.
- Level diagrams [9] [10]: Esta técnica se basa en diagramas de nivel, donde cada objetivo y parámetro de diseño se representa en un diagrama separado. Esta representación se basa en dos ideas: 1) clasificación de los individuos según su proximidad a los puntos ideales medidos con una norma específica (norma 1, norma 2 o norma infinito) con los objetivos normalizados; 2) sincronización de objetivos y parámetros de diseño. De esta forma, en cada diagrama de nivel se representan el valor objetivo o parámetro en cuestión en el eje de abscisas y su valor de norma empleada en el eje de ordenadas. Además, gracias a la sincronización, seleccionando una o varias alternativas en cualquier diagrama, se selecciona automáticamente su representación en el resto de ellos. Para poder establecer las preferencias del diseñador y ayudar a la toma de decisiones, los diagramas de nivel pueden ser en color.

## 2.6 Métricas

En esta sección se presentan una serie de métricas empleadas para validar y cuantificar las prestaciones de un MOEA. Los conjuntos  $P_Q^*$  y  $P_{Q,\epsilon}^*$  no son únicos, y por ello, es importante cuantificar sus prestaciones. Los MOEAs tienen una componente aleatoria y por este motivo será necesario añadir una medida estadística, intentando evitar en gran medida este fenómeno. En la actualidad, las métricas empleadas [45] [82] para validar los MOEAs van dirigidas a medir: convergencia, diversidad y convergencia-diversidad.

### 2.6.1 Métricas de convergencia

Las métricas de convergencia miden el grado de proximidad en la distancia entre un conjunto de soluciones (frente de Pareto obtenido  $f(P_Q^*)$ ) y el frente de Pareto verdadero  $f(P_Q)$ . Diferentes métricas para medir la convergencia han sido estudiadas por autores en la literatura [122] [45] [82]. Una de las más utilizadas es la distancia generacional ( $GD$  [102] [103]). Para el empleo de esta métrica es fundamental conocer una buena aproximación a  $P_Q$ . Teniendo una buena aproximación de  $P_Q$  y un  $P_Q^*$  con  $n_{P_Q}$  y  $n_{P_Q^*}$  individuos respectivamente, y  $m$  objetivos de diseño, se calcula la  $GD$  como:

$$GD(f(P_Q^*), f(P_Q)) = \frac{(\sum_{j=1}^{n_{P_Q^*}} d(\mathbf{x}^j)^p)^{1/p}}{n_{P_Q^*}}, \quad \mathbf{x}^j \in P_Q^* \quad (2.3)$$

donde

$$d(\mathbf{x}^j) = \min_{\mathbf{x}^1 \in P_Q} \sqrt{\sum_{i=1}^m (f_i(\mathbf{x}^1) - f_i(\mathbf{x}^j))^2} \quad (2.4)$$

y  $p$  es un parámetro entero.

La  $GD$  promedia la distancia mínima, sobre el espacio de objetivos, entre cada solución de  $P_Q^*$  y el conjunto  $P_Q$ . El objetivo es que este valor sea lo más pequeño posible, siendo  $GD = 0$  cuando  $P_{Q,n}^* \subseteq P_{Q,n}$ . Pese a que esta métrica está aceptada por muchos autores, algunos no la consideran. Una posible explicación a este hecho es su estrategia de promedio. Veamos un ejemplo. Supongamos que tenemos un conjunto de Pareto  $P_Q$  y un conjunto discreto  $P_{1Q}^*$  que dispone de un único individuo con distancia euclídea mínima 5 a  $P_Q$  en el espacio de objetivos ( $d(\mathbf{x}^j) = 5$ ). Así, con  $p = 2$ ,  $GD(f(P_{1Q}^*), f(P_Q)) = \sqrt{5^2}/1 = 5$ . Ahora, disponemos de otra población obtenida  $P_{2Q}^*$  con 5 individuos idénticos al de  $P_{1Q}^*$  (con  $d(\mathbf{x}^j) = 5$ ). En este caso, con  $p = 2$ , se tiene  $GD(f(P_{2Q}^*), f(P_Q)) = \sqrt{5^2 + 5^2 + 5^2 + 5^2 + 5^2}/5 = 2.236$ . Según este indicador,  $P_{2Q}^*$  mejora en convergencia a  $P_{1Q}^*$ . Sin embargo, aparentemente  $P_{2Q}^*$  no tiene mejor convergencia que  $P_{1Q}^*$ . De hecho,  $GD$  tiende a 0 con una muestra de infinitos individuos en el conjunto  $P_Q^*$ . Para evitar esta problemática, una variante de esta métrica ( $GD_p$ ) ha sido propuesta en [90]:

$$GD_p(f(P_Q^*), f(P_Q)) = \left( \frac{\sum_{j=1}^{n_{P_Q^*}} d(\mathbf{x}^j)^p}{n_{P_Q^*}} \right)^{1/p}, \quad \mathbf{x}^j \in P_Q^* \quad (2.5)$$

$GD_p$  consigue corregir este efecto indeseado y por ello parece una métrica más justa. Suponiendo el ejemplo descrito anteriormente, se tiene que  $GD_p(f(P_{1Q}^*), f(P_Q)) = GD_p(f(P_{2Q}^*), f(P_Q))$ , lo cual parece un resultado más razonable.

### 2.6.2 Métrica diversidad

Las métricas de diversidad miden la distribución de las soluciones en el espacio de objetivos. Diferentes métricas con este objetivo han sido estudiadas por autores en la literatura [82]. Una de ellas es la métrica de diversidad ( $\Delta'$  [22]). Para el empleo de esta métrica no es necesario conocer una buena aproximación al frente de Pareto verdadero. Teniendo un conjunto de soluciones a analizar  $P_Q^*$  con  $n_{P_Q^*}$  individuos, se calcula la métrica  $\Delta'$  de la siguiente forma:

$$\Delta'(f(P_Q^*)) = \frac{\sum_{i=1}^{n_{P_Q^*}-1} d_i - \bar{d}}{(n_{P_Q^*} - 1)} \quad (2.6)$$

donde  $d_i$  es la distancia euclídea entre soluciones consecutivas de  $P_Q^*$  y  $\bar{d}$  es el promedio de  $d_i$ . Si todos los pares de soluciones consecutivas comparten la misma distancia ( $d_i = \bar{d}$ ),  $\Delta'(f(P_Q^*)) = 0$ , y  $P_Q^*$  tiene una distribución perfecta en el espacio de objetivos. Para encontrar soluciones consecutivas, el requisito previo de esta métrica es ordenar las soluciones de  $P_Q^*$  según su valor en el espacio de objetivos. Esta métrica tiene como ventaja fundamental que no es necesario conocer una buena aproximación del conjunto  $P_Q$ .

$\Delta'$  está diseñado para su utilización en el espacio de objetivos. Sin embargo, también es útil conocer la diversidad en el espacio de parámetros. El frente de Pareto puede tener soluciones en diferentes vecindarios (en el espacio de parámetros) y es importante que el conjunto obtenido logre detectarlos. Este problema es todavía mayor al considerar soluciones casi-óptimas ya que las posibilidades de encontrar nuevos vecindarios aumenta. Una métrica que mida la diversidad en el espacio de parámetros puede darnos una medida de cuantos individuos significativamente diferentes, pertenecientes a  $P_{Q,\epsilon}$ , han sido encontrados, lo que es una información muy valiosa para el diseñador.

### 2.6.3 Métricas convergencia-diversidad

Cada vez está más extendido el uso de métricas que miden la convergencia y diversidad simultáneamente. Estas métricas permiten medir en un solo indicador las dos características del conjunto obtenido más importantes para el diseñador. Una de las métricas más empleadas con este propósito es el hipervolumen (HV [121] [120]). Esta métrica proporciona una medida del volumen que está dominado por el conjunto de soluciones obtenidas ( $P_Q^*$ ) [4] [106]. El HV es, por lo tanto, el área comprendida entre un conjunto de referencia ( $R$ ) y  $P_Q^*$  (aproximación al frente de Pareto). Cuanto más cercano sea  $P_Q^*$  al frente de Pareto verdadero mayor será el tamaño del área dominada por  $P_Q^*$ , y por lo tanto, mayor será el valor del HV. Del mismo modo, valores altos de HV indican que el conjunto  $P_Q^*$  está más uniformemente distribuido (mayor área dominada) en el espacio de objetivos. De esta forma, HV proporciona una medida de convergencia y diversidad simultáneamente.

$$HV(P_Q^*, R) = \text{volumen} \left( \bigcup_{i=1}^{|P_Q^*|} v_i \right) \quad (2.7)$$

donde  $v_i$  es el hipercubo construido entre el conjunto de referencia y cada solución del conjunto obtenido.

Sin embargo, HV tiene tres inconvenientes. En primer lugar, HV puede ser engañoso si el conjunto aproximado no es convexo. En segundo lugar, el coste computacional para el cálculo de esta métrica es elevado. Este coste crece exponencialmente con el número de objetivos, por lo tanto, se vuelve inabordable con un gran número de ellos. En tercer lugar, el HV solo es aplicable en el espacio de objetivos, no siendo posible medir la diversidad en el espacio de parámetros (importante para el diseñador como se detalla en la sección anterior).

Otra de las métricas más empleadas para medir la convergencia y diversidad de un conjunto encontrado es una variante de la GD, la distancia generacional invertida (IGD [122] [50]):

$$IGD(f(P_Q^*), f(P_Q)) = \frac{(\sum_{j=1}^{n_{P_Q}} d(\mathbf{x}^j)^p)^{1/p}}{n_{P_Q}}, \quad \mathbf{x}^j \in P_Q \quad (2.8)$$

donde

$$d(\mathbf{x}^j) = \min_{\mathbf{x}^1 \in P_Q^*} \sqrt{\sum_{i=1}^m (f_i(\mathbf{x}^1) - f_i(\mathbf{x}^j))^2} \quad (2.9)$$

y  $p$  es un parámetro entero.

Esta métrica generalmente se utiliza en el espacio de objetivos, siendo además posible su utilización en el espacio de parámetros. Un valor bajo de  $IGD$  indica una buena convergencia-diversidad del conjunto obtenido. Sin embargo, esta métrica presenta la misma problemática que  $GD$ , antes mencionada. Supongamos que tenemos un conjunto obtenido  $P_Q^*$  y dos conjuntos objetivos  $P_{1Q}$  y  $P_{2Q}$ , el primero de ellos con una discretización más fina en el espacio de objetivos. Así, con  $p = 2$ ,  $IGD(f(P_Q^*), f(P_{1Q})) < IGD(f(P_Q^*), f(P_{2Q}))$  pese a que el conjunto  $P_Q^*$  es idéntico. De hecho,  $IGD$  tiende a 0 con una muestra de infinitos individuos en el conjunto  $P_Q$ . Para evitar este efecto indeseado se propone en [90] el indicador  $IGD_p$  (análogo a  $GD_p$ ):

$$IGD_p = \left( \frac{\sum_{j=1}^{n_{P_Q}} d(\mathbf{x}^j)^p}{n_{P_Q}} \right)^{1/p}, \quad \mathbf{x}^j \in P_Q \quad (2.10)$$

Suponiendo el ejemplo anterior,  $IGD(f(P_Q^*), f(P_{1Q})) = IGD(f(P_Q^*), f(P_{2Q}))$ , y en consecuencia, parece una métrica más justa.

Otra métrica utilizada en la literatura para medir la convergencia y diversidad de un conjunto obtenido es la distancia *Hausdorff* ( $d_H$  [89] [92]).  $d_H$  es una medida de la distancia a la que se encuentran dos subconjuntos. Por lo tanto, disponiendo de una buena aproximación a  $P_Q$ , la  $d_H$  se puede emplear para medir la calidad de un conjunto discreto obtenido  $P_Q^*$  en el espacio de objetivos y parámetros. La  $d_H$  se mide como:

$$d_H(P_Q^*, P_Q) = \max(\text{dist}(P_Q^*, P_Q), \text{dist}(P_Q, P_Q^*)) \quad (2.11)$$

siendo la distancia  $\text{dist}(A, B)$  entre dos conjuntos:

$$\text{dist}(A, B) = \sup_{a_i \in A} \text{dist}(a_i, B) \quad (2.12)$$

siendo la distancia  $\text{dist}(a_i, B)$  entre un individuo y un conjunto:

$$\text{dist}(a_i, B) = \inf_{b_i \in B} d_\infty(a_i, b_i) \quad (2.13)$$

siendo la distancia  $d_\infty(a_i, b_i)$  entre dos individuos:

$$d_\infty(a_i, b_i) = \max_{i=1, \dots, m} |a_i - b_i| \quad (2.14)$$

Si el valor de  $d_H$  es 0, significa que  $P_Q^* = P_Q$ , ya que  $\text{dist}(P_Q^*, P_Q) = 0$  y  $\text{dist}(P_Q, P_Q^*) = 0$  ( $P_Q^* \subseteq P_Q$  y  $P_Q \subseteq P_Q^*$ ). Por ello, se deduce que los valores pequeños de  $d_H$  indican buenas aproximaciones de  $P_Q^*$ . En caso de que el valor de  $d_H$  no sea pequeño, se requiere un análisis más profundo. La  $\text{dist}(P_Q^*, P_Q)$  mide la convergencia hacia  $P_Q$ , mientras que  $\text{dist}(P_Q, P_Q^*)$  mide la diversidad de  $P_Q^*$  a lo largo de  $P_Q$ . Sin embargo la  $d_H$  tiene dos inconvenientes: 1) la consideración de los outliers (valores atípicos que pueden dar una falsa impresión sobre la calidad de la aproximación); 2) solo tiene en cuenta la mayor distancia entre conjuntos, lo que puede ser un valor engañoso al no considerar el resto del conjunto. Supongamos que medimos  $d_H$  sobre el espacio de parámetros de un MOP. Se tiene un conjunto de Pareto ( $P_Q$ ) que se compone de 3 individuos equidistantes  $\mathbf{x}^1$ ,  $\mathbf{x}^2$  y  $\mathbf{x}^3$ . Ahora se obtienen dos conjuntos a analizar  $P_{1Q}^*$  y  $P_{2Q}^*$ .  $P_{1Q}^*$  tiene dos individuos  $\mathbf{x}^1$  y  $\mathbf{x}^2$ .  $P_{2Q}^*$  tiene un único individuo  $\mathbf{x}^2$ . En este caso,  $d_H(P_{1Q}^*, P_Q) = d_H(P_{2Q}^*, P_Q)$  pese a que, lógicamente,  $P_{1Q}^*$  es una aproximación mejor a  $P_Q$ .

Para evitar esta problemática, surge un nuevo indicador utilizado en la literatura, la distancia media de Hausdorff  $\Delta_p(P_Q^*, P_Q)$  [90]:

$$\Delta_p(P_Q^*, P_Q) := \max(GD_p(P_Q^*, P_Q), IGD_p(P_Q^*, P_Q)) \quad (2.15)$$

$\Delta_p$  obtiene las ventajas de las métricas  $GD_p$  e  $IGD_p$ . Además, esta métrica no se distorsiona ante valores atípicos (comunes en los MOEAs) debido al promedio de las distancias entre los conjuntos comparados. Suponiendo el ejemplo anterior  $\Delta_p(f(P_{1Q}^*), f(P_Q)) < \Delta_p(f(P_{2Q}^*), f(P_Q))$  lo que indica, como es lógico, una mejor aproximación del conjunto  $P_{1Q}^*$ . Esta métrica será empleada para la evaluación del algoritmo diseñado (nevMOGA) en este trabajo, utilizando para ello  $p = 2$ .

## Capítulo 3

# Algoritmo nevMOGA



## 3.1 Introducción

En el capítulo anterior se han introducido los conceptos básicos relacionados con la optimización multiobjetivo clásica y la obtención del conjunto de soluciones casi-óptimas. Además, también se han presentado los algoritmos evolutivos existentes destinados a la localización de alternativas casi-óptimas. Estos algoritmos discretizan solo en función del espacio de objetivos. Esta metodología puede ser adecuada para algunas aplicaciones, sin embargo, una solución casi-óptima puede ser eliminada sin tener en cuenta su ubicación en el espacio de parámetros, y en consecuencia, las soluciones casi-óptimas que tienen características distintas podrían ser descartadas. Esto provocaría una pérdida de soluciones que consideramos especialmente relevantes por la característica de ser diferentes.

Por este motivo, en este capítulo se presenta un nuevo algoritmo: nevMOGA. nevMOGA surge de la necesidad de garantizar que las alternativas casi-óptimas que son significativamente diferentes (no vecinas) a las que le dominan no son descartadas. Este algoritmo considera el conjunto de alternativas óptimas y casi-óptimas, discretizando el espacio de parámetros y objetivos simultáneamente.

Este capítulo se organiza de la siguiente manera: En primer lugar, en la sección 3.2, se presenta el nuevo conjunto de interés que pretende contener las soluciones casi-óptimas no dominadas en su vecindad (soluciones potencialmente útiles). Para obtener un número manejable de soluciones, nevMOGA debe discretizar el conjunto definido. Esta discretización se introduce en la sección 3.3. En tercer lugar, en la sección 3.4, se describe el algoritmo nevMOGA, capaz de obtener el conjunto de interés definido anteriormente. Por último, en la sección 3.5, se evalúa nevMOGA sobre dos *benchmarks*. En esta evaluación, nevMOGA se compara con otros dos algoritmos: búsqueda aleatoria y búsqueda exhaustiva. Esta comparación se realiza utilizando la distancia media de *Hausdorff*  $\Delta_p$  (ver sección 2.6.3), un indicador que mide la convergencia y diversidad de los conjuntos obtenidos. Cada algoritmo comparado se evaluará 50 veces para obtener una validación estadística, evitando en la medida de lo posible el efecto aleatorio de los algoritmos en cuestión.

nevMOGA se ha diseñado para caracterizar las soluciones casi-óptimas no dominadas en su vecindad, a diferencia de otros algoritmos evolutivos existentes (ver sección 2.4) que solo caracterizan soluciones óptimas y casi-óptimas. Por este motivo, no es posible realizar una comparación coherente entre ellos.

### 3.2 Nuevo conjunto de interés

Diferentes autores detallan la importancia de considerar el conjunto de alternativas casi-óptimas. Sin embargo, obtenerlas todas conlleva un número excesivo de soluciones, lo que provoca dos inconvenientes: ralentiza el proceso de optimización y complica en exceso la etapa de decisión. Además, es importante no descuidar las alternativas que siendo similares en el espacio de objetivos tengan parámetros significativamente diferentes. Para conseguir este propósito, es necesario discretizar el conjunto de interés teniendo en cuenta también el espacio de parámetros. Por lo tanto, el nuevo conjunto de interés debe: (1) obtener un número manejable de soluciones (de modo que se puedan evitar los dos problemas mencionados anteriormente); (2) no descuidar la diversidad existente en las características de las soluciones casi-óptimas. Las soluciones que, bajo nuestro criterio, cumplen estas características, son las alternativas que son significativamente diferentes a las que le dominan, es decir, las alternativas no dominadas en su vecindad (zona de parámetros). Estas alternativas, que llamaremos potencialmente útiles, aportan la información más relevante al diseñador sin aumentar en exceso el número de ellas. Para definir el conjunto de interés es necesario introducir los conceptos de vecindad,  $n$ -dominancia, y el conjunto de interés  $P_{Q,n}$  en su imagen en el espacio de parámetros y objetivos.

**Definición 3.1** (*vecindad*): Definido  $\mathbf{n} = [n_1, \dots, n_k]$  como la máxima distancia entre soluciones vecinas. Dos vectores solución  $\mathbf{x}^1$  y  $\mathbf{x}^2$  son soluciones vecinas si  $|x_i^1 - x_i^2| < n_i$  para todo  $i \in [1, \dots, k]$ , siendo  $k$  el número de variables de decisión. Esto se denota como  $\mathbf{x}^1 =_n \mathbf{x}^2$ .

**Definición 3.2** ( $n$ -dominancia): Un vector solución  $\mathbf{x}^1$  es  $n$ -dominado por otro vector solución  $\mathbf{x}^2$  si, siendo ellas soluciones vecinas (definición 3.1),  $\mathbf{x}^2 \preceq \mathbf{x}^1$ . Esto se denota como  $\mathbf{x}^2 \preceq_n \mathbf{x}^1$ .

**Definición 3.3** (*Conjunto  $P_{Q,n}$* ):  $P_{Q,n}$  es el conjunto de soluciones pertenecientes a  $P_{Q,\epsilon}$  que son no  $n$ -dominadas por otra solución de  $P_{Q,\epsilon}$ :

$$P_{Q,n} := \{\mathbf{x} \in P_{Q,\epsilon} \mid \nexists \mathbf{x}' \in P_{Q,\epsilon} : \mathbf{x}' \preceq_n \mathbf{x}\}$$

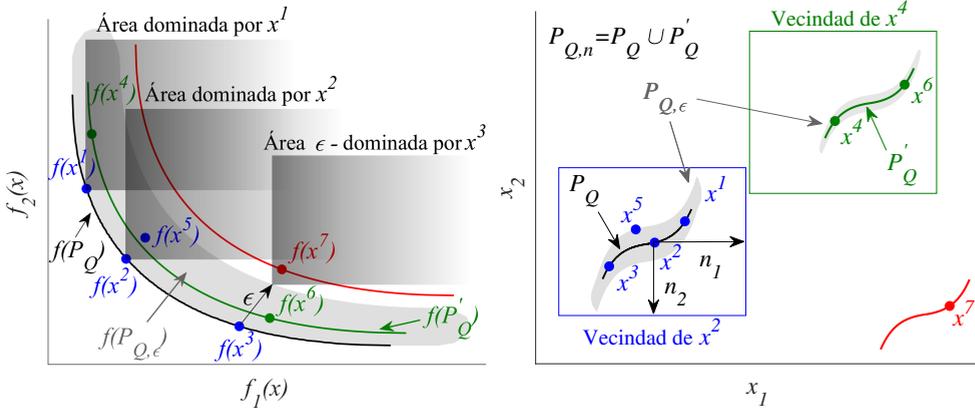
**Definición 3.4** (*Conjunto  $\mathbf{f}(P_{Q,n})$* ): Dado un conjunto de soluciones óptimas y casi-óptimas no dominadas en su vecindad  $P_{Q,n}$ ,  $\mathbf{f}(P_{Q,n})$  es definido como:

$$\mathbf{f}(P_{Q,n}) := \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in P_{Q,n}\}$$

Del mismo modo que con los conjuntos  $P_Q$  y  $P_{Q,\epsilon}$ , habitualmente es suficiente con determinar un conjunto discreto  $P_{Q,n}^* \subset P_{Q,n}$ .  $P_{Q,n}^*$  no es un conjunto único y deberá caracterizar adecuadamente a  $P_{Q,n}$ .

En la figura 3.1 se observa un caso particular de un MOP con dos objetivos de diseño  $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x})]$ , dos variables de decisión  $\mathbf{x} = [x_1 \ x_2]$ , una máxima degradación de rendimiento aceptable  $\epsilon$  y una vecindad dada  $\mathbf{n} = [n_1 \ n_2]$ . Se tienen 7 posibles soluciones al MOP ( $\mathbf{x}^1$  a  $\mathbf{x}^7$ ). En primer lugar,  $\mathbf{x}^1$ ,  $\mathbf{x}^2$  y  $\mathbf{x}^3$  forman parte del conjunto  $P_Q$ , y por lo tanto, son soluciones óptimas. En segundo lugar,  $\mathbf{x}^4$  es una solución casi-óptima dominada por  $\mathbf{x}^1$  pero no  $n$ -dominada.  $\mathbf{x}^1$  y  $\mathbf{x}^4$  son alternativas con parámetros significativamente diferentes y con un rendimiento similar, y por lo tanto, ambas soluciones aportan nueva información relevante al diseñador, pese a que una de ellas este dominada. Es posible que  $\mathbf{x}^4$  presente alguna mejora significativa respecto a  $\mathbf{x}^1$  en cualquier características no incluida (o agregada) en los objetivos de diseño. El diseñador deberá analizar estas características para elegir la alternativa más adecuada a sus especificaciones. Por ello, creemos que  $\mathbf{x}^4$  es una alternativa potencialmente útil. Lo mismo ocurre con  $\mathbf{x}^6$ . En tercer lugar,  $\mathbf{x}^5$  es una alternativa dominada por una solución vecina ( $\mathbf{x}^2$ ), y por ello, es una solución prescindible bajo nuestro enfoque. El diseñador siempre elegirá  $\mathbf{x}^2$ , ya que teniendo ambas características similares (debido a que sus parámetros son también similares), es mejor en los objetivos de diseño. Por último,  $\mathbf{x}^7$  es una solución  $\epsilon$ -dominada por  $\mathbf{x}^3$ , y en consecuencia, no es una alternativa casi-óptima. Con todo ello, el conjunto  $P_{Q,n}$  es el formado por los conjuntos  $P_Q$  y  $P'_Q$ . El conjunto discreto  $P_{Q,n}^*$  obtenido, en este ejemplo concreto, es el formado por las alternativas  $\mathbf{x}^1$ ,  $\mathbf{x}^2$ ,  $\mathbf{x}^3$ ,  $\mathbf{x}^4$  y  $\mathbf{x}^6$ . Lógicamente no es la única posibilidad, ni probablemente la mejor, de caracterizar  $P_{Q,n}$ .

El conjunto  $P_{Q,n}$  depende directamente de la máxima degradación de rendimiento aceptable  $\epsilon$  y del tamaño de la vecindad  $\mathbf{n}$ . Dado que en la mayoría de los problemas, los objetivos tienen un sentido físico, es posible decidir a priori cuánto rendimiento estamos dispuestos a perder, y en consecuencia, definir  $\epsilon$ . De manera similar, cuando las variables de decisión tienen un sentido físico, la elección del parámetro  $\mathbf{n}$  puede ser trivial. Configurando  $\mathbf{n}$ , el diseñador establece en qué medida dos soluciones se consideran similares. Si las variables de decisión no tienen un sentido físico, es posible que la elección de  $\mathbf{n}$  sea compleja. En este caso, es posible seguir un procedimiento simple (ver figura 3.2) que nos ayudará en esta elección. En primer lugar, se elige una solución de referencia  $\mathbf{x}^R$ . En segundo lugar, se define, en el espacio de objetivos, un rectángulo con centro en  $\mathbf{f}(\mathbf{x}^R)$  y ancho y alto dos veces cada componente del vector  $\epsilon$ . Por ejemplo, en este caso (ver figura 3.2) tenemos un rectángulo de



**Figura 3.1:** Ejemplo de un MOP con dos objetivos y dos variables de decisión. Conjunto de soluciones óptimas y casi-óptimas no dominadas en su vecindad  $P_{Q,n}$  formado por los conjuntos  $P_Q$  y  $P'_Q$ . El vector  $\epsilon = [\epsilon_1 \ \epsilon_2]$  esta formado por  $\epsilon_1$  y  $\epsilon_2$  (degradación aceptable para  $f_1$  y  $f_2$  respectivamente).

ancho dos veces  $\epsilon_1$  (máxima degradación de rendimiento aceptable sobre  $f_1$ ) y alto dos veces  $\epsilon_2$  (máxima degradación de rendimiento aceptable sobre  $f_2$ ). Posteriormente, se aumenta y disminuye cada componente del vector solución  $\mathbf{x}^R$  de forma independiente. El proceso, en cada dirección (aumento o disminución de cada componente de  $\mathbf{x}^R$ ), finaliza cuando una solución abandona el rectángulo definido previamente en el espacio de objetivos. Finalmente, cada elemento del vector vecindad  $\mathbf{n}$  se establece como la distancia mínima entre  $\mathbf{x}^R$  y la primera solución (una en cada dirección) que no se encuentra dentro del rectángulo mencionado ( $n_1$  y  $n_2$  en la figura 3.2). A través de este sencillo procedimiento, logramos cuantificar qué excursión es necesaria en el espacio de búsqueda para obtener cambios significativos en el espacio objetivo, y de este modo, podemos definir  $\mathbf{n}$  cuando no hay otro criterio disponible.

### 3.3 Discretización del nuevo conjunto de interés

El conjunto  $P_{Q,n}$  puede contener infinitas soluciones, por ello, es necesario discretizarlo y obtener un conjunto finito de soluciones ( $P_{Q,n}^*$ ) que lo caracterice adecuadamente. Para llevar a cabo esta tarea, los conceptos de *box*, *box* dominancia y  $n$  - *box* dominancia deben definirse de antemano.

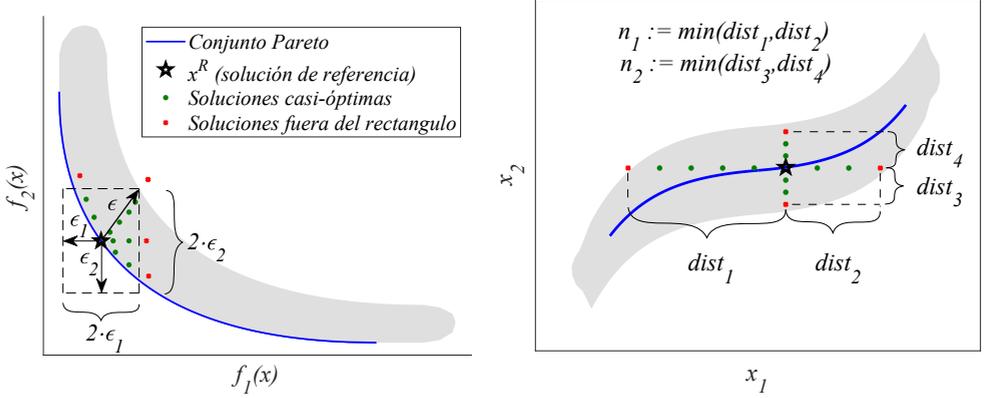


Figura 3.2: Procedimiento para la determinación de la vecindad  $n$ .

**Definición 3.5** (Box [37]): Definido  $\delta_i = \frac{f_i^{max} - f_i^{min}}{n\_box_i}$ , siendo  $n\_box = [n\_box_1, \dots, n\_box_m]$ , para todo  $i \in [1, \dots, m]$ , siendo  $m$  el número de objetivos, el número de divisiones en cada dimensión, se define el box de un vector solución  $\mathbf{x}$  como el vector  $\mathbf{box}(\mathbf{x}) = [box_1(\mathbf{x}), \dots, box_m(\mathbf{x})]$ , donde:

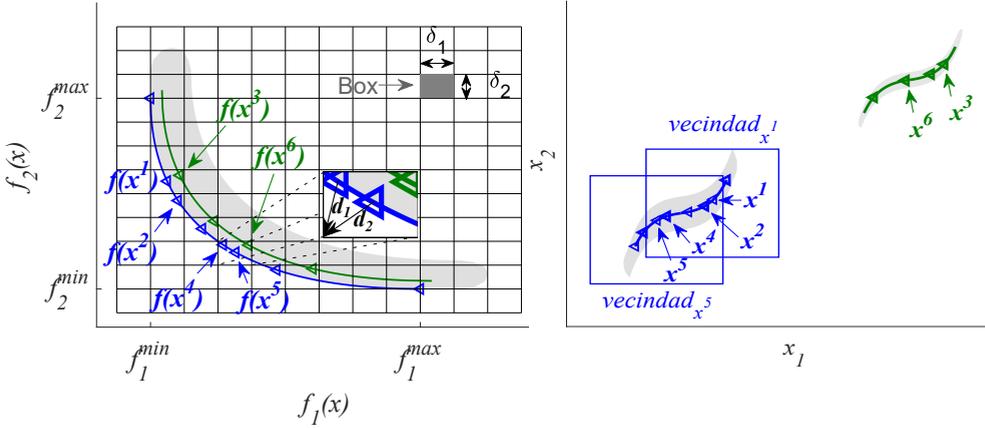
$$box_i = \lfloor \frac{f_i(\mathbf{x}) - f_i^{min}}{\delta_i} \rfloor \quad \forall i \in [1, \dots, m]$$

para  $\delta_i > 0$ , y donde  $f_i^{max}$  y  $f_i^{min}$  son los valores máximos y mínimos de  $\mathbf{f}(\mathbf{x})$ .

**Definición 3.6** (Box dominancia [37]): Dados dos vectores solución  $\mathbf{x}^1$  y  $\mathbf{x}^2$  cuyos boxes son  $\mathbf{box}(\mathbf{x}^1)$  y  $\mathbf{box}(\mathbf{x}^2)$  respectivamente, se dice que  $\mathbf{x}^1$  box domina a  $\mathbf{x}^2$ ,  $\mathbf{box}(\mathbf{x}^1) \preceq \mathbf{box}(\mathbf{x}^2)$ , si  $box_i(\mathbf{x}^1) \leq box_i(\mathbf{x}^2)$  para todo  $i \in [1, \dots, m]$  y  $box_j(\mathbf{x}^1) < box_j(\mathbf{x}^2)$  para al menos un  $j$ ,  $j \in [1, \dots, m]$ .

**Definición 3.7** ( $n$ -box dominancia): Dados dos vectores solución  $\mathbf{x}^1$  y  $\mathbf{x}^2$  cuyos boxes son  $\mathbf{box}(\mathbf{x}^1)$  y  $\mathbf{box}(\mathbf{x}^2)$  respectivamente, se dice que  $\mathbf{x}^1$   $n$ -box domina a  $\mathbf{x}^2$  si, siendo ambas soluciones vecinas ( $\mathbf{x}^1 =_n \mathbf{x}^2$ ),  $\mathbf{box}(\mathbf{x}^1) \preceq \mathbf{box}(\mathbf{x}^2)$ . Esto se denota como  $\mathbf{box}(\mathbf{x}^1) \preceq_n \mathbf{box}(\mathbf{x}^2)$ .

Cada solución es ubicada en un  $box$ . Una solución es  $box$  dominada si existe otra solución en un  $box$  que domine al suyo (mejor o igual en todos los objetivos de diseño y mejor en al menos uno). El algoritmo solo mantiene soluciones que no están  $n$ -box dominadas, es decir, solo admite soluciones que no están  $box$



**Figura 3.3:** Un ejemplo de discretización de un MOP mediante nevMOGA.

dominadas por una solución vecina. Además, un *box* no puede contener dos o más soluciones vecinas. Cuando se encuentren dos o más soluciones vecinas en el mismo *box*, nevMOGA elegirá la solución más cercana al vértice ideal (inferior izquierdo, consulte la figura 3.3,  $d_1$  y  $d_2$ ), descartando el resto de ellas. En la figura 3.3 se muestra un ejemplo de la discretización del conjunto  $P_{Q,n}$ . El espacio de objetivos está distribuido en *boxes*.  $x^1$  está  $n - box$  dominada (es decir, *box* dominada por una solución vecina) por  $x^2$  y, por lo tanto, es descartada.  $x^3$  está *box* dominada pero no  $n - box$  dominada (ya que  $x^2$  y  $x^3$  no son soluciones vecinas), y en consecuencia, es una alternativa potencialmente útil.  $x^4$ ,  $x^5$  y  $x^6$  están ubicadas en el mismo *box*. Dos soluciones vecinas no pueden estar en el mismo *box*, y por ello,  $x^4$  o  $x^5$  (vecinas) deben ser descartadas. En este caso, se elimina  $x^4$  al tener una distancia mayor al vértice ideal ( $d_2 < d_1$ ).  $x^6$  es una alternativa potencialmente útil porque siendo una solución casi-óptima, no está  $n - box$  dominada y no pertenece al vecindario del resto de alternativas ubicadas en el mismo *box* ( $x^4$  y  $x^5$ ). Los conjuntos discretos  $P_Q^*$  y  $P_{Q,n}^*$ , obtenidos por nevMOGA, tienen un número máximo de soluciones (en el peor de los casos)  $|P_Q^*|$  y  $|P_{Q,n}^*|$  respectivamente (ver ecuaciones 3.1 y 3.2).

$$|P_Q^*| \leq \frac{\prod_{i=1}^m n_{-box}}{n_{-box_{max}}} \quad (3.1)$$

$$|P_{Q,n}^*| \leq \frac{\prod_{i=1}^m (\lceil \frac{\epsilon_i}{\delta_i} \rceil + n\_box_i)}{m} \prod_{j=1}^k \frac{\bar{x}_j - x_j}{n_i} \quad (3.2)$$

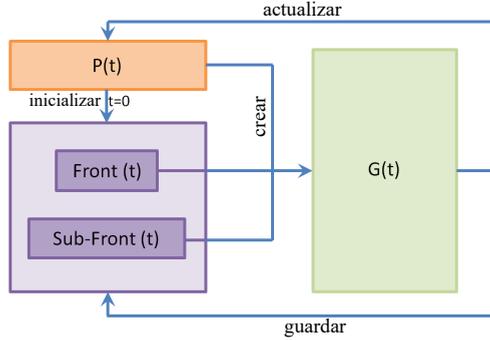
$$i = 1$$

### 3.4 Descripción de nevMOGA

nevMOGA se basa en el algoritmo evolutivo ev-MOGA [38]. nevMOGA a diferencia de ev-MOGA obtiene, además del conjunto de soluciones óptimas, las alternativas casi-óptimas no dominadas en su vecindad, almacenándolas en una población adicional. nevMOGA es un algoritmo elitista que utiliza operadores de cruce y mutación para generar nuevos individuos en cada iteración. Este algoritmo tiene cuatro poblaciones (figura 3.4):

1.  $P(t)$  es la población encargada de explorar el espacio de búsqueda, con el objetivo de obtener soluciones de  $P_{Q,n}$  (y no solo de  $P_Q$ ), manteniendo la diversidad en el conjunto de soluciones, ayudando en el proceso de creación de nuevos individuos. El número de individuos en esta población es constante e igual a  $Nind_P$ .
2.  $Front(t)$  es el archivo donde se almacena  $P_Q^*$ , es decir, una aproximación discreta del frente de Pareto. El tamaño de esta población ( $|P_Q^*|$ ) varía, pero está acotado dependiendo del número de *boxes* previamente definidos por el usuario mediante el parámetro  $n\_box$  (ver ecuación 3.1).
3.  $Sub-Front(t)$  es el archivo donde se almacena  $P_{Q,n}^* \setminus P_Q^*$ , es decir, las soluciones casi-óptimas no dominadas en su vecindad. Su tamaño ( $|P_{Q,n}^*|$ ) es variable pero acotado, dependiendo del número de *boxes* y vecindades (ver ecuación 3.2).
4.  $G(t)$  es una población auxiliar donde se almacenan los nuevos individuos generados por el algoritmo en la iteración actual. En primer lugar, se generan 4 individuos, repitiendo el proceso hasta completar la población. Por ello, el número de individuos de esta población ( $Nind_G$ ) debe ser múltiplo de 4.

Una vez introducidos los archivos  $Front(t)$  y  $Sub-Front(t)$ , se pueden establecer las condiciones que debe cumplir una solución para poder ingresar en ellos:



**Figura 3.4:** Estructura de nevMOGA formada por cuatro poblaciones.

**Definición 3.8** (*Inclusión de  $\mathbf{x}$  en  $Front(t)$* ): Dado un vector solución  $\mathbf{x}$  y un archivo  $Front(t)$ ,  $\mathbf{x}$  será incluido en  $Front(t)$  si y solo si:

$$\nexists \mathbf{x}^* \in Front(t) : (\mathbf{box}(\mathbf{x}^*) \preceq \mathbf{box}(\mathbf{x})) \wedge (\mathbf{box}(\mathbf{x}^*) = \mathbf{box}(\mathbf{x}) \wedge dist(\mathbf{x}^*) \leq dist(\mathbf{x}))$$

donde  $dist(\mathbf{z})$  es la distancia desde  $\mathbf{z}$  hasta el vértice ideal (inferior izquierdo) del  $\mathbf{box}$  al que pertenece (ver figura 3.3,  $\mathbf{d}_1$  y  $\mathbf{d}_2$ ). Además, si  $\mathbf{x}$  es incluido en  $Front(t)$ , entonces todas las soluciones  $\mathbf{x}^*$  que cumplan la siguiente condición serán eliminadas de  $Front(t)$ :

$$(\mathbf{box}(\mathbf{x}) \preceq \mathbf{box}(\mathbf{x}^*)) \vee (\mathbf{box}(\mathbf{x}) = \mathbf{box}(\mathbf{x}^*) \wedge dist(\mathbf{x}) \leq dist(\mathbf{x}^*))$$

Si  $\mathbf{x}$  no es incluida en  $Front(t)$  debe estudiarse su inclusión en  $Sub-Front(t)$ . Lo mismo ocurre con las soluciones  $\mathbf{x}^*$  eliminadas de  $Front(t)$ .

**Definición 3.9** (*Inclusión de  $\mathbf{x}$  en  $Sub-Front(t)$* ): Dada una solución  $\mathbf{x}$  tal que  $\mathbf{x} \notin Front(t)$ ,  $\mathbf{x}$  será incluida en  $Sub-Front(t)$  si y solo si:

$$\begin{aligned} \nexists \mathbf{x}^* \in Front(t) : \mathbf{x}^* \preceq_{-\epsilon} \mathbf{x} \wedge \nexists \mathbf{x}^* \in Front(t) \cup Sub-Front(t) : \\ (\mathbf{box}(\mathbf{x}^*) \preceq_n \mathbf{box}(\mathbf{x})) \wedge (\mathbf{box}(\mathbf{x}^*) =_n \mathbf{box}(\mathbf{x}) \wedge dist(\mathbf{x}^*) \leq dist(\mathbf{x})) \end{aligned}$$

Además, si  $\mathbf{x}$  es incluida en  $Sub-Front(t)$ , entonces todas las soluciones  $\mathbf{x}^*$  que cumplan la siguiente condición serán eliminadas de  $Sub-Front(t)$ :

$$(\mathbf{box}(\mathbf{x}) \preceq_n \mathbf{box}(\mathbf{x}^*)) \vee (\mathbf{box}(\mathbf{x}) =_n \mathbf{box}(\mathbf{x}^*) \wedge \text{dist}(\mathbf{x}) \leq \text{dist}(\mathbf{x}^*))$$

Para mantener la diversidad de soluciones en  $P_{Q,n}^*$ , es necesario que la población  $P(t)$  esté adecuadamente distribuida. Supongamos que tenemos dos soluciones en  $P(t)$   $\mathbf{x}^1$  y  $\mathbf{x}^2$ .  $\mathbf{x}^1$  es una solución que se encuentra en un vecindario aislado *vecindario*<sub>1</sub>, sin otra alternativa con parámetros similares.  $\mathbf{x}^2$  es una alternativa que se encuentra en un vecindario muy poblado *vecindario*<sub>2</sub>, es decir, que tiene un gran número de soluciones con parámetros similares en  $P(t)$ . Si se elige aleatoriamente una solución de  $P(t)$  para ser evolucionada, es más probable elegir una solución del *vecindario*<sub>2</sub> que del *vecindario*<sub>1</sub> al tener un mayor número de soluciones. Así, después de un gran número de iteraciones el *vecindario*<sub>1</sub> tiende a estar despoblado (o eliminado), mientras que el *vecindario*<sub>2</sub> tendrá una gran evolución. Para obtener un conjunto adecuadamente distribuido en ambos vecindarios es necesario ayudar a los vecindarios/soluciones menos poblados. Con este propósito, se ordenan permanentemente las soluciones de  $P(t)$  mediante el uso del indicador denominado *niche count* (ver definición 3.10). Este indicador cuantifica la densidad de las soluciones, siendo preferidas las soluciones poco pobladas. De este modo,  $\mathbf{x}^1$  tendrá un menor valor del indicador *niche count*, y en consecuencia, durante el proceso de selección, tendrá mayor probabilidad de ser elegida para generar nuevos individuos que  $\mathbf{x}^2$ . Del mismo modo, en el proceso de sustitución es menos probable sustituir a  $\mathbf{x}^1$ , debido a su valor inferior del indicador *niche count*, que a  $\mathbf{x}^2$  (ver algoritmo 5).

**Definición 3.10** (*niche count* [86]):

$$\text{niche}_i = \sum_{j=1}^{N_{ind_p}} sh(d_{ij})$$

donde  $sh(d_{ij})$  es la función de *sharing*, que proporciona una medida de lo similares que son dos elementos en una población, y se define por:

$$sh(d_{ij}) = \begin{cases} \sum_{j=1}^k \left(1 - \frac{|x_j^1 - x_j^2|}{n_j}\right) & \text{if } \mathbf{x}^1 =_n \mathbf{x}^2 \\ 0 & \text{else} \end{cases}$$

En el algoritmo 5 se observa el pseudocódigo principal de nevMOGA. Las líneas 3, 6, 8, 10, 13, 14 y 15 del pseudocódigo anterior constituyen los cambios que se han agregado a ev-MOGA, el algoritmo en el que se basa nevMOGA. A continuación, se explican los pasos más importantes de nevMOGA:

---

**Algoritmo 5** Pseudocódigo principal.

---

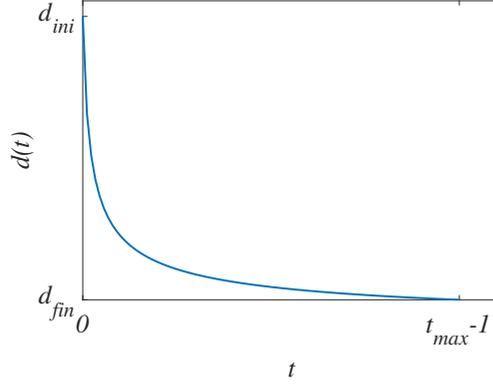
```

1:  $t := 0$ ;
2:  $\text{Front}(t) := \emptyset$ ;
3:  $\text{Sub-Front}(t) := \emptyset$ ;
4: Crear población inicial  $P(t)$  aleatoriamente
5: Calcular  $f(\mathbf{x}) \forall \mathbf{x} \in P(t)$ 
6: Ordenar población  $P(t)$  en función del indicador niche count ▷ Definición 3.10
7: Analizar inclusión de los individuos de  $P(t)$  en  $\text{Front}(t)$  ▷ Algoritmo 6
8: Analizar inclusión de los individuos de  $P(t) \notin \text{Front}(t)$  en  $\text{Sub-Front}(t)$  ▷ Algoritmo 7
9: for  $t := 1$ : Numero de iteraciones do
10:   Crear población  $G(t)$  ▷ Algoritmo 8
11:   Calcular  $f(\mathbf{x}) \forall \mathbf{x} \in G(t)$ 
12:   Analizar inclusión de los individuos de  $G(t)$  en  $\text{Front}(t)$  ▷ Algoritmo 6
13:   Analizar inclusión de los individuos de  $G(t) \notin \text{Front}(t)$  en  $\text{Sub-Front}(t)$  ▷ Algoritmo 7
14:   Actualización  $P(t)$  con los individuos de  $G(t)$  ▷ Algoritmo 9
15:   Ordenar población  $P(t)$ 
16: end for

```

---

- Líneas 7 y 12 (algoritmo 6): Aquí se determina si un nuevo individuo será incluido en  $\text{Front}(t)$ . Para esto, el individuo debe cumplir con la definición 3.8. Si el nuevo individuo se incluye en  $\text{Front}(t)$  debe analizarse si su inclusión elimina a otros individuos de  $\text{Front}(t)$  (en cuyo caso se establecerá si deben incluirse en  $\text{Sub-Front}(t)$ ) y  $\text{Sub-Front}(t)$ .
- Líneas 8 y 13 (algoritmo 7): Aquí se determina si un nuevo individuo, que no se incluyó en  $\text{Front}(t)$ , tiene que ser incluido en  $\text{Sub-Front}(t)$ . Para esto, el individuo debe cumplir con la definición 3.9. Si el nuevo individuo se incluye en el  $\text{Sub-Front}(t)$  debe analizarse si su inclusión elimina a otros individuos del archivo  $\text{Sub-Front}(t)$ .
- Línea 10 (algoritmo 8): La población  $G(t)$  se crea mediante el siguiente procedimiento:
  1. Se seleccionan dos individuos al azar, uno de  $\text{Front}(t)$  ( $\mathbf{x}^F$ ) (donde cualquier individuo tiene la misma probabilidad de ser elegido) y otro de  $P(t)$  ( $\mathbf{x}^P$ ) (donde cada uno tiene una probabilidad diferente de ser elegido). Las soluciones de  $P(t)$  se ordenan según su indicador *niche count* (de menor a mayor) y el individuo  $\mathbf{x}^P$  se elige de acuerdo con una distribución exponencial, lo que hace que sea más probable que se elija una solución con un valor más bajo de *niche count* (menos poblada). Este procedimiento favorece la distribución uniforme de las soluciones.



**Figura 3.5:** Gráfico de  $d(t)$  desde  $t = 0$  a  $t = t_{max} - 1$ . Los parámetros  $d_{ini}$  y  $d_{fin}$  son los valores iniciales y finales de  $d(t)$ .

2. Se genera un número aleatorio  $u \in [0, 1]$ . Si  $u > P_{c/m}$  (probabilidad de cruce/mutación), entonces se realizará un cruce (paso 3), de lo contrario se realizará una mutación (paso 4).
3.  $\mathbf{x}^P$  y  $\mathbf{x}^F$  se cruzan por medio de un operador de cruce intermedio extendido, que produce dos nuevos individuos  $\mathbf{x}^{h1}$  y  $\mathbf{x}^{h2}$ :

$$\begin{aligned} x_i^{h1} &= \alpha_i(t)x_i^P + (1 - \alpha_i(t))x_i^F \\ x_i^{h2} &= (1 - \alpha_i(t))x_i^P + (\alpha_i(t))x_i^F \end{aligned}$$

El parámetro  $\alpha_i(t)$  es un valor aleatorio distribuido uniformemente que pertenece al intervalo  $[-d(t), 1+d(t)]$  y  $d(t)$  es un parámetro que se ajusta mediante el uso de una función de disminución exponencial, como en *simulated annealing* [37]:

$$d(t) = \frac{d_{ini}}{\sqrt{1 + \left( \left( \frac{d_{ini}}{d_{fin}} \right)^2 - 1 \right) \frac{t}{t_{max}-1}}}, \quad d_{ini} < d_{fin}$$

La figura 3.5 muestra  $d(t)$  desde  $t = 0$  a  $t = t_{max} - 1$ , donde  $d(0) = d_{ini}$  y  $d(t_{max} - 1) = d_{fin}$ .

4.  $\mathbf{x}^P$  y  $\mathbf{x}^F$  se mutan usando una mutación aleatoria con distribución gaussiana:

$$\begin{aligned} x_i^{h1} &= x_i^P + N(0, \beta1_i(t)) \\ x_i^{h2} &= x_i^F + N(0, \beta2_i(t)) \end{aligned}$$

donde las variaciones  $\beta 1_i(t)$  y  $\beta 2_i(t)$  se expresan como un porcentaje de  $x_{imax} - x_{imin}$ . Estas variaciones se calculan mediante una función que es similar a la utilizada anteriormente para el parámetro  $d(t)$ :

$$\beta(t) = \frac{\beta_{ini}}{\sqrt{1 + \left( \left( \frac{\beta_{ini}}{\beta_{fin}} \right)^2 - 1 \right) \frac{t}{(t_{max} - 1)}}}$$

5.  $\mathbf{x}^{h1}$  y  $\mathbf{x}^{h2}$  son incluidas en  $G(t)$ .

Una vez finalizados estos cinco pasos, se repite el proceso de nuevo con una solución  $\mathbf{x}^{SF}$ , del archivo Sub-Front(t), en lugar de  $\mathbf{x}^F$  (solución del archivo Front(t)). Este proceso se repite hasta alcanzar el tamaño definido del archivo  $G(t)$ .

- Línea 14 (algoritmo 9): En la actualización de  $P(t)$  se determina si un nuevo individuo de  $G(t)$  ( $\mathbf{x}^G$ ) sustituye a cualquier otro individuo existente en  $P(t)$ . La búsqueda del individuo que se va a sustituir ( $\mathbf{x}^P$ ) comienza desde un individuo elegido por una distribución exponencial, inversa a la utilizada en la elección de  $\mathbf{x}^P$  para la creación de  $G(t)$  (Línea 10, algoritmo 8) y, por lo tanto, ahora será más probable elegir soluciones más pobladas. Esto tiene como objetivo lograr una distribución más uniforme de las soluciones. Durante esta actualización, pueden ocurrir tres casos diferentes:

1. El nuevo individuo  $\mathbf{x}^G$  es  $\epsilon$ -dominado por algún miembro de  $Front(t)$ . En este caso, se buscará sustituir un individuo  $\mathbf{x}^P$  dominado por  $\mathbf{x}^G$ .
2. El nuevo individuo  $\mathbf{x}^G$  no está  $\epsilon$ -dominado por ningún miembro de  $Front(t)$  y existe un individuo  $\mathbf{x}^P$  dominado por  $\mathbf{x}^G$ , que es  $\epsilon$ -dominado por algún miembro de  $Front(t)$ . En este caso será sustituido.
3. El nuevo individuo  $\mathbf{x}^G$  no está  $\epsilon$ -dominado por ningún miembro de  $Front(t)$  y no existe un individuo  $\mathbf{x}^P$  dominado por  $\mathbf{x}^G$  que es  $\epsilon$ -dominado por algún miembro de  $Front(t)$ . En este caso, se elige la solución inicial  $\mathbf{x}^P$ . Si  $\mathbf{x}^P$  y  $\mathbf{x}^G$  son soluciones vecinas, se buscará una solución  $n$ -dominada por  $\mathbf{x}^G$  para sustituirla. Si no lo son, se buscará una solución  $n$ -dominada por  $\mathbf{x}^P$  para sustituirla. En este último caso, si no se encuentra esa solución,  $\mathbf{x}^G$  sustituirá aleatoriamente a una solución de la vecindad de  $\mathbf{x}^P$ .

**Algoritmo 6** Inclusión de  $\mathbf{x}$  en  $\text{Front}(t)$ .

▷ Definición 3.8

---

```

1: if  $\exists \mathbf{x}^* \in \text{Front}(t) : (\mathbf{box}(\mathbf{x}^*) \preceq \mathbf{box}(\mathbf{x}))$  then return
2: else if  $\exists \mathbf{x}^* \in \text{Front}(t) : (\mathbf{box}(\mathbf{x}^*) = \mathbf{box}(\mathbf{x}) \ \& \ \text{dist}(\mathbf{x}^*) \leq \text{dist}(\mathbf{x}))$  then return
3: end if
4: Incluir  $\mathbf{x}$  en  $\text{Front}(t)$ ;
5: if  $\exists \mathbf{x}^* \in \text{Front}(t) : (\mathbf{box}(\mathbf{x}) \preceq \mathbf{box}(\mathbf{x}^*))$  then                                ▷ Elimina soluciones no óptimas
6:   Eliminar  $\mathbf{x}^*$  de  $\text{Front}(t)$ 
7: else if  $\exists \mathbf{x}^* \in \text{Front}(t) : (\mathbf{box}(\mathbf{x}) = \mathbf{box}(\mathbf{x}^*) \ \& \ \text{dist}(\mathbf{x}) \leq \text{dist}(\mathbf{x}^*))$  then
8:   Eliminar  $\mathbf{x}^*$  de  $\text{Front}(t)$ 
9: end if
10: if  $\exists \mathbf{x}^* \in \text{Sub-Front}(t) : \mathbf{x} \preceq_{-\epsilon} \mathbf{x}^*$  then                                ▷ Elimina soluciones no casi-óptimas
11:   Eliminar  $\mathbf{x}^*$  de  $\text{Sub-Front}(t)$ 
12: end if

```

---

**Algoritmo 7** Inclusión de  $\mathbf{x}$  en  $\text{Sub-Front}(t)$ .

▷ Definición 3.9

---

```

1: if  $\exists \mathbf{x}^* \in \text{Front}(t) : \mathbf{x}^* \preceq_{-\epsilon} \mathbf{x}$  then return
2: else if  $\exists \mathbf{x}^* \in \text{Front}(t) \cup \text{Sub-Front}(t) : (\mathbf{box}(\mathbf{x}^*) \preceq_n \mathbf{box}(\mathbf{x}))$  then return
3: else if  $\exists \mathbf{x}^* \in \text{Front}(t) : (\mathbf{box}(\mathbf{x}^*) =_n \mathbf{box}(\mathbf{x}) \ \& \ \text{dist}(\mathbf{x}^*) \leq \text{dist}(\mathbf{x}))$  then return
4: end if
5: Incluir  $\mathbf{x}$  en  $\text{Sub-Front}(t)$ ;
6: if  $\exists \mathbf{x}^* \in \text{Sub-Front}(t) : (\mathbf{box}(\mathbf{x}) \preceq_n \mathbf{box}(\mathbf{x}^*))$  then                                ▷ Eliminar soluciones no casi-óptimas
7:   Eliminar  $\mathbf{x}^*$  de  $\text{Sub-Front}(t)$ 
8: else if  $\exists \mathbf{x}^* \in \text{Sub-Front}(t) : (\mathbf{box}(\mathbf{x}) =_n \mathbf{box}(\mathbf{x}^*) \ \& \ \text{dist}(\mathbf{x}) \leq \text{dist}(\mathbf{x}^*))$  then
9:   Eliminar  $\mathbf{x}^*$  de  $\text{Sub-Front}(t)$ 
10: end if

```

---

**Algoritmo 8** Crear población  $G(t)$ .

---

```

1: Elegir  $\mathbf{x}^F \in \text{Front}(t)$                                 ▷ Selección aleatoria
2: Elegir  $\mathbf{x}^{P1} \in P(t)$                                 ▷ Mediante una distribución exponencial ( $\mu = N \text{ind}_P / 10$ )
3:  $u := \text{random}(1)$ 
4: if  $u > P_{c/m}$  then                                    ▷  $P_{c/m}$  es la probabilidad de cruce y mutación
5:    $\mathbf{x}^{G1}$  y  $\mathbf{x}^{G2}$  son obtenidos mediante el cruce de  $\mathbf{x}^{P1}$  y  $\mathbf{x}^F$ 
6: else
7:    $\mathbf{x}^{G1}$  y  $\mathbf{x}^{G2}$  son obtenidos mediante la mutación de  $\mathbf{x}^{P1}$  y  $\mathbf{x}^F$ 
8: end if
9: Elegir  $\mathbf{x}^{SF} \in \text{Sub-Front}(t)$                                 ▷ Selección aleatoria
10: Elegir  $\mathbf{x}^{P2} \in P(t)$                                     ▷ Mediante una distribución exponencial ( $\mu = N \text{ind}_P / 10$ )
11:  $u := \text{random}(1)$ 
12: if  $u > P_{c/m}$  then
13:    $\mathbf{x}^{G3}$  y  $\mathbf{x}^{G4}$  son obtenidos mediante el cruce de  $\mathbf{x}^{P2}$  and  $\mathbf{x}^{SF}$ 
14: else
15:    $\mathbf{x}^{G3}$  and  $\mathbf{x}^{G4}$  son obtenidos mediante la mutación de  $\mathbf{x}^{P2}$  y  $\mathbf{x}^{SF}$ 
16: end if

```

---

---

**Algoritmo 9** Actualización de  $P(t)$  con los individuos de  $G(t)$ .

---

```

1: Elegir  $x^{P_{Initial}} \in P(t)$   $\triangleright$  Mediante una distribución exponencial ( $\mu = N_{ind_P}/10$ )
2: if  $\nexists x' \in \text{Front}(t) : x' \preceq_{-\epsilon} x^G$  then
3:   if  $\exists x^* \in P(t) : \exists x'' \in \text{Front}(t) : x'' \preceq_{-\epsilon} x^* \ \& \ x^G \preceq x^*$  then  $\triangleright$  Búsqueda  $x^*$  iniciada en  $x^{P_{Initial}}$ 
4:      $x^* := x^G$  return
5:   else
6:      $X_N := \forall x^* \in P(t) : x^{P_{Initial}} =_n x^*$ 
7:     if  $x^G =_n x^{P_{Initial}}$  then
8:       if  $\exists x^* \in X_N : x^G \preceq x^*$  then  $\triangleright$  La búsqueda de  $x^*$  comienza por  $x^{P_{Initial}}$ 
9:          $x^* := x^G$  return
10:      end if
11:     else
12:       if  $\exists x_1^* \ \& \ x_2^* \in X_N : x_1^* \preceq x_2^*$  then
13:          $x_2^* := x^G$  return
14:       end if
15:     end if
16:   end if
17:    $x^{P_{Random}} = \text{Random}(X_N)$   $\triangleright$  Selección aleatoria de un individuo de  $X_N$ 
18:    $x^{P_{Random}} := x^G$ 
19: else
20:   if  $\exists x^* \in P(t) : x^G \preceq x^*$  then  $\triangleright$  La búsqueda de  $x^*$  comienza por  $x^{P_{Initial}}$ 
21:      $x^* := x^G$ 
22:   end if
23: end if

```

---

### 3.5 Evaluación

En esta sección se evalúa nevMOGA sobre dos *benchmarks* planteados. Estos resultados se analizarán mediante un indicador que mide convergencia y diversidad de conjuntos (distancia media de *Hausdorff*  $\Delta_p$ ). Aunque existen algoritmos evolutivos que buscan el conjunto de soluciones casi-óptimas [91] [92], ninguno de ellos tiene el mismo conjunto de intereses que nevMOGA ( $P_{Q,n}$ ). Por esta razón, nevMOGA no se puede comparar directamente con ninguno de ellos. Sin embargo, nevMOGA sí puede compararse con otros dos algoritmos: búsqueda aleatoria y exhaustiva. En esta sección se realiza esta comparación. Cada algoritmo evalúa la función objetivo el mismo número de veces. Cada algoritmo se ejecuta 50 veces sobre cada *benchmark*, evitando, en la medida de lo posible, el efecto aleatorio de los mismos. Para el caso de búsqueda exhaustiva, las soluciones evaluadas se distribuyen uniformemente sobre el espacio de búsqueda, formando un *hipergrid*. Este *hipergrid* se desplaza ligeramente en cada eje para obtener diferentes conjuntos en cada ejecución.

### 3.5.1 Métrica

Para comparar el rendimiento de los tres algoritmos se utiliza la distancia media de *Hausdorff*  $\Delta_p$ . Mediante este indicador se puede medir el rendimiento entre un conjunto discreto obtenido  $P_{Q,n}^*$ , con  $n_{P_{Q,n}^*}$  individuos, y un conjunto objetivo  $\overline{P_{Q,n}^*}$  (discretización muy precisa de  $P_{Q,n}$ ), con  $n_{\overline{P_{Q,n}^*}}$  individuos. Este rendimiento se puede medir tanto en el espacio de objetivos como en el espacio de parámetros. De esta forma, es posible medir la convergencia hacia el frente de Pareto y la diversidad de soluciones en el conjunto de Pareto. Es deseable que este indicador tenga el menor valor posible. Idealmente, cuando  $P_{Q,n}^* = \overline{P_{Q,n}^*}$ , entonces  $\Delta_p = 0$ .

$$\Delta_p(P_{Q,n}^*, \overline{P_{Q,n}^*}) := \max(GD_p(P_{Q,n}^*, \overline{P_{Q,n}^*}), IGD_p(P_{Q,n}^*, \overline{P_{Q,n}^*})) \quad (3.3)$$

donde  $GD_p$  y  $IGD_p$  están definidos en la sección 2.6.

$\Delta_p$  obtiene las ventajas de las métricas  $GD_p$  y  $IGD_p$ . Por un lado,  $GD_p$ , a diferencia de  $GD$ , no está influida por el tamaño del conjunto obtenido  $P_{Q,n}^*$ , es decir, su rendimiento es independiente del tamaño del conjunto a comparar. Por otro lado,  $IGD_p$ , a diferencia de  $IGD$ , no está influenciado por la discretización del conjunto objetivo, es decir, su rendimiento es independiente del tamaño del conjunto objetivo  $\overline{P_{Q,n}^*}$ . Además, esta métrica no se distorsiona ante valores atípicos (comunes en los MOEAs) debido al promedio de las distancias entre los conjuntos comparados.

### 3.5.2 Benchmark 1

El primer *benchmark* considerado en este trabajo corresponde al ejemplo presentado en [92]:

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x})] \quad (3.4)$$

$$\begin{aligned} f_1(\mathbf{x}) &= (x_1 - t_1(c + 2a) + a)^2 + (x_2 - t_2b)^2 + \delta_t \\ f_2(\mathbf{x}) &= (x_1 - t_1(c + 2a) - a)^2 + (x_2 - t_2b)^2 + \delta_t \end{aligned} \quad (3.5)$$

donde

$$t_1 = \text{sgn}(x_1) \min \left( \left\lceil \frac{|x_1| - a - c/2}{2a + c} \right\rceil, 1 \right), \quad t_2 = \text{sgn}(x_2) \min \left( \left\lceil \frac{|x_2| - b/2}{b} \right\rceil, 1 \right)$$

y

$$\delta_t = \begin{cases} 0 & \text{for } t_1 = 0 \text{ and } t_2 = 0 \\ 0.1 & \text{else} \end{cases}$$

sujeto a:

$$\begin{aligned} \underline{\mathbf{x}} &= [-8 \quad -8] \\ \bar{\mathbf{x}} &= [8 \quad 8] \end{aligned}$$

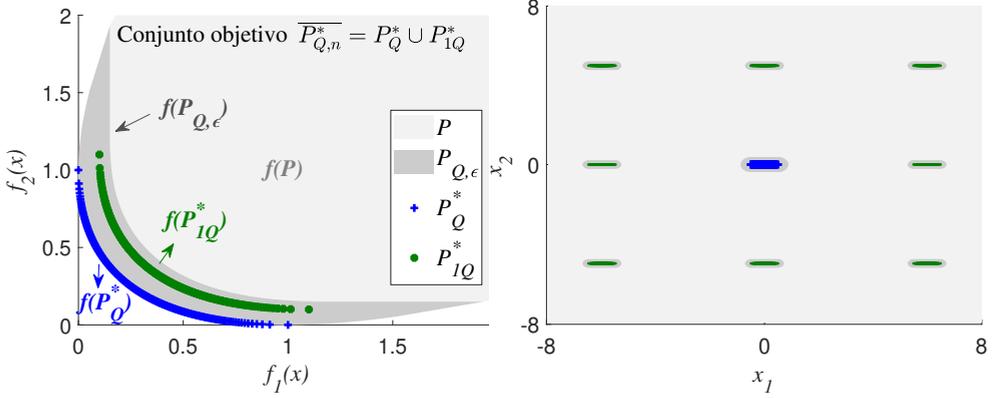
siendo  $a = 0.5$ ,  $b = 5$  y  $c = 5$ , este MOP tiene un conjunto de Pareto:

$$P_{0,0} = [-0.5, 0.5] \times \{0\} = P_Q \quad (3.6)$$

y ocho conjuntos locales de Pareto:

$$\begin{aligned} P_{-1,-1} &= [-6.5, -5.5] \times \{-5\} \\ P_{0,-1} &= [-0.5, 0.5] \times \{-5\} \\ P_{1,-1} &= [5.5, 6.5] \times \{-5\} \\ P_{-1,0} &= [-6.5, -5.5] \times \{0\} \\ P_{1,0} &= [5.5, 6.5] \times \{0\} \\ P_{-1,1} &= [-6.5, -5.5] \times \{5\} \\ P_{0,1} &= [-0.5, 0.5] \times \{5\} \\ P_{1,1} &= [5.5, 6.5] \times \{5\} \end{aligned} \quad (3.7)$$

En este *benchmark*, dado que se conocen el conjunto de Pareto y los conjuntos de Pareto locales, podemos establecer los parámetros  $\epsilon$  y  $\mathbf{n}$  más fácilmente. Aquí, establecemos  $\epsilon = [0.15 \ 0.15]$  para que los ocho conjuntos locales descritos se consideren soluciones casi-óptimas. Obtenemos  $\mathbf{n}$  aplicando el procedimiento descrito en la sección 3.2 con una solución de referencia  $\mathbf{x}^R = [0 \ 0]$  y el parámetro  $\epsilon$  establecido previamente, resultando  $\mathbf{n} = [0.13 \ 0.38]$ . Una vez que se han determinado estos parámetros, el conjunto  $P_{Q,n}$  es el formado por:



**Figura 3.6:** Conjunto objetivo  $\overline{P_{Q,n}^*}$ , formado por  $P_Q^* \cup P_{1Q}^*$ . Este conjunto está formado por nueve subconjuntos, uno de ellos contiene las soluciones óptimas, el resto el conjunto de soluciones casi-óptimas.

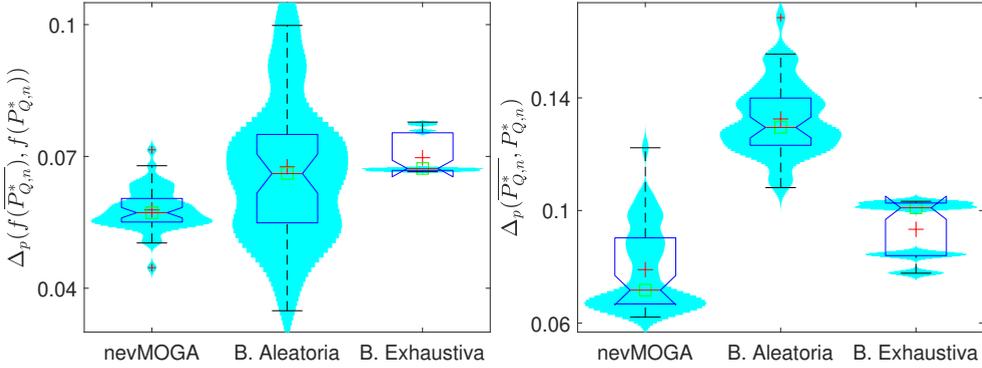
$$P_{Q,n} := P_{0,0} \cup P_{-1,-1} \cup P_{0,-1} \cup P_{1,-1} \cup P_{-1,0} \cup P_{1,0} \cup P_{-1,1} \cup P_{0,1} \cup P_{1,1} \quad (3.8)$$

Como se ha mencionado en la sección anterior, es necesario obtener un conjunto objetivo  $\overline{P_{Q,n}^*}$  (discretización muy fina de  $P_{Q,n}$ ) para poder comparar los conjuntos obtenidos por los algoritmos a evaluar. Este conjunto se muestra en la figura 3.6. Para el cálculo de cada conjunto  $P_{Q,n}^*$ , obtenido mediante nevMOGA, se ha configurado nevMOGA con los siguientes parámetros:

$$\begin{aligned} P_{c/m} &= 0.2 \\ \mathbf{n\_box} &= [15 \ 15] \\ Nind_G &= 8 \\ Nind_P &= 600 \\ \text{Iteraciones} &= 550 \end{aligned}$$

el resto de parámetros toman el valor por defecto sugerido en [37] para el algoritmo original (ev-MOGA).

La figura 3.7 y la tabla 3.1 muestran los resultados estadísticos de los tres algoritmos (cada uno se ejecuta 50 veces en este *benchmark*) que se han comparado utilizando  $\Delta_p$ . Cada algoritmo evalúa la función objetivo 5000 veces en cada



**Figura 3.7:** Comparación de los algoritmos mediante Boxplots, sobre el *benchmark* 1. A la izquierda,  $\Delta_p(f(P_{Q,n}^*), f(\overline{P_{Q,n}^*}))$  mide la aproximación de los conjuntos obtenidos hacia el conjunto objetivo en el espacio de objetivos  $f(\overline{P_{Q,n}^*})$ . A la derecha,  $\Delta_p(P_{Q,n}^*, \overline{P_{Q,n}^*})$  mide la aproximación de los conjuntos obtenidos hacia el conjunto objetivo  $\overline{P_{Q,n}^*}$ .

ejecución. Los conjuntos obtenidos (uno por cada ejecución) por los tres algoritmos se comparan con el conjunto objetivo  $\overline{P_{Q,n}^*}$ . Esta comparación se puede llevar a cabo en el espacio de objetivos y parámetros de forma independiente. A la izquierda de la figura 3.7 se muestran los resultados sobre el espacio de objetivos  $\Delta_p(\mathbf{f}(P_{Q,n}^*), \mathbf{f}(\overline{P_{Q,n}^*}))$ , mientras que a la derecha se observan los resultados sobre el espacio de parámetros  $\Delta_p(P_{Q,n}^*, \overline{P_{Q,n}^*})$ , utilizando Boxplots y estimación de densidad. Los valores de  $\Delta_p$  producidos por nevMOGA en ambos espacios son buenos (cerca de 0) y mejores que los proporcionados por sus competidores (búsqueda aleatoria y exhaustiva). Para la búsqueda aleatoria, la calidad de aproximación al frente varía enormemente. Para la búsqueda exhaustiva, los resultados muestran menos variabilidad, aunque se comporta peor que nevMOGA en ambos espacios. Además, una prueba de comparación múltiple ([100], [24]) lleva a la conclusión de que los resultados de los tres algoritmos son significativamente diferentes, bajo  $\Delta_p$ , en ambos espacios. Por estas razones, se puede concluir que nevMOGA supera significativamente al resto de algoritmos comparados sobre el *benchmark* 1.

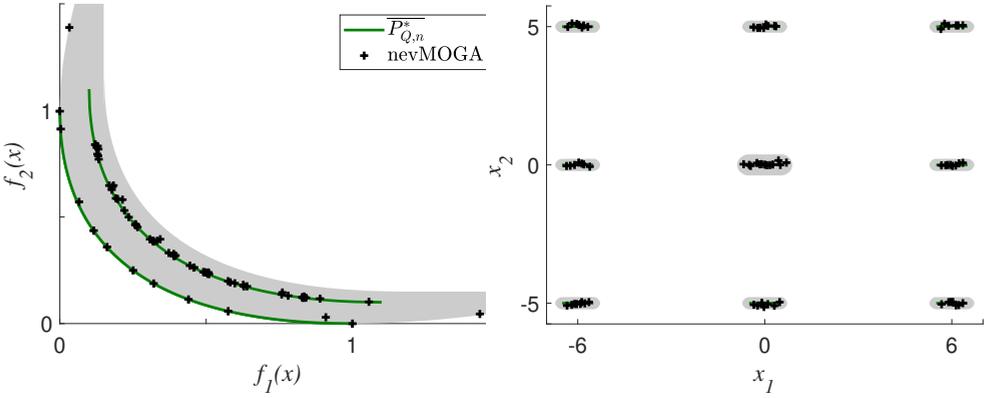
Para comparar gráficamente los tres algoritmos, se ha elegido un conjunto obtenido  $P_{Q,n}^*$  (de las 50 ejecuciones) de cada uno de ellos. El conjunto  $P_{Q,n}^*$  escogido es el que tiene valores de  $\Delta_p(P_{Q,n}^*, \overline{P_{Q,n}^*})$  y  $\Delta_p(\mathbf{f}(P_{Q,n}^*), \mathbf{f}(\overline{P_{Q,n}^*}))$  más cercanos a los valores promedio. En las figuras 3.8, 3.9 y 3.10 se observan estos conjuntos comparados con el conjunto objetivo  $\overline{P_{Q,n}^*}$ . El conjunto obtenido por nevMOGA tiene 70 soluciones, mientras que los obtenidos por la búsque-

**Tabla 3.1:** Valores numéricos de los resultados estadísticos presentados en la figura 3.7.

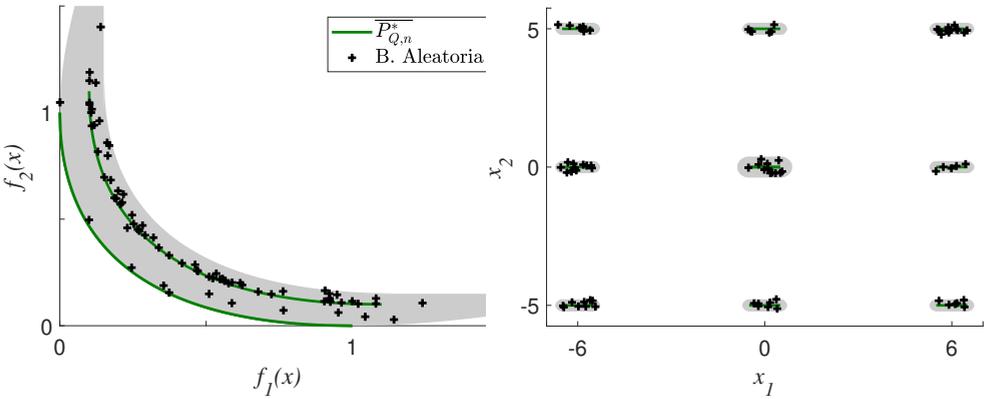
	$\Delta_p(f(P_{Q,n}^*), f(\overline{P_{Q,n}^*}))$		
	nevMOGA	Aleatoria	Exhaustiva
Máximo	0.0715	0.121	0.0778
Q3	0.0604	0.0750	0.0754
Media	0.0578	0.0676	0.0697
Mediana	0.0572	0.0661	0.0673
Q1	0.0551	0.0549	0.0668
Mínimo	0.0447	0.0349	0.0665
	$\Delta_p(P_{Q,n}^*, \overline{P_{Q,n}^*})$		
	nevMOGA	Aleatoria	Exhaustiva
Máximo	0.122	0.169	0.103
Q3	0.0903	0.140	0.103
Media	0.0790	0.133	0.0933
Mediana	0.0717	0.130	0.101
Q1	0.0668	0.123	0.0840
Mínimo	0.0621	0.108	0.0777

da aleatoria y exhaustiva tienen 78 y 48 soluciones respectivamente. La gran cantidad de soluciones encontradas por nevMOGA además de los resultados mostrados demuestran que nevMOGA es capaz de caracterizar mejor, el conjunto objetivo  $\overline{P_{Q,n}^*}$ , que sus competidores sobre *benchmark* 1. Este resultado, más el hecho de que nevMOGA alcanza valores más bajos de la métrica  $\Delta_p$ , nos lleva a la conclusión de que nevMOGA logra un mejor desempeño general en este *benchmark*.

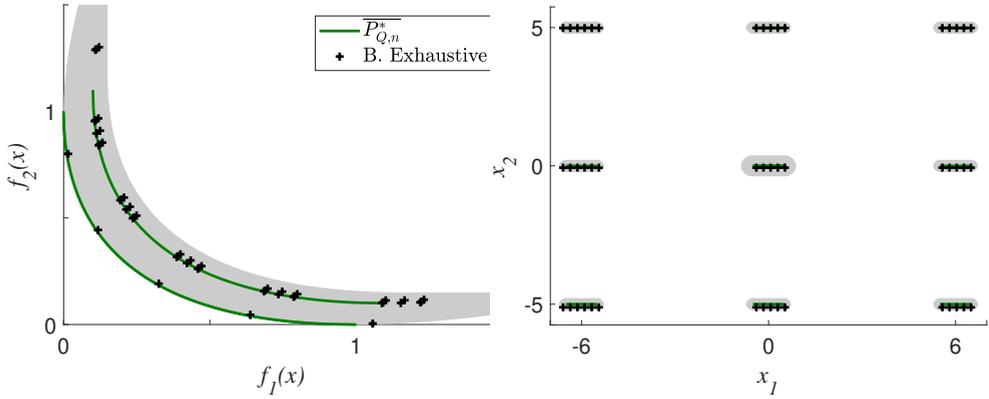
Este *benchmark* también ha sido resuelto por [92], donde se caracteriza el conjunto  $P_{Q,\epsilon}$ , mediante el algoritmo  $P_{Q,\epsilon}$ -MOEA. En este trabajo mencionado, se observa como  $P_{Q,\epsilon}$ -MOEA obtiene un número significativamente mayor de soluciones (450 soluciones aproximadamente) que nevMOGA (70 soluciones). nevMOGA consigue caracterizar todos los vecindarios existentes (con soluciones óptimas y casi-óptimas) obteniendo un número significativamente menor de soluciones. Por lo tanto, se ha cumplido el objetivo principal de nevMOGA: encontrar las soluciones que proporcionan la información más relevante sin aumentar en exceso el número de ellas. De esta forma se ha evitado ralentizar el algoritmo y se ha simplificado la etapa de toma de decisiones.



**Figura 3.8:** Comparación de un conjunto  $P_{Q,n}^*$ , obtenido mediante nevMOGA, y el conjunto objetivo  $\overline{P_{Q,n}^*}$ , en el espacio de parámetros (derecha) y objetivos (izquierda), sobre el *benchmark 1*.



**Figura 3.9:** Comparación de un conjunto  $P_{Q,n}^*$ , obtenido mediante búsqueda aleatoria, y el conjunto objetivo  $\overline{P_{Q,n}^*}$ , en el espacio de parámetros (derecha) y objetivos (izquierda), sobre el *benchmark 1*.



**Figura 3.10:** Comparación de un conjunto  $P_{Q,n}^*$ , obtenido mediante búsqueda exhaustiva, y el conjunto objetivo  $\overline{P_{Q,n}^*}$ , en el espacio de parámetros (derecha) y objetivos (izquierda), sobre el *benchmark* 1.

### 3.5.3 Benchmark 2

El segundo *benchmark* es una adaptación del *benchmark Modified Rastrigin* [51]. Este *benchmark* se ha modificado para convertirlo en un MOP con un conjunto óptimo y soluciones casi-óptimas en diferentes vecindarios. Esta es su formulación:

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x})] \quad (3.9)$$

con

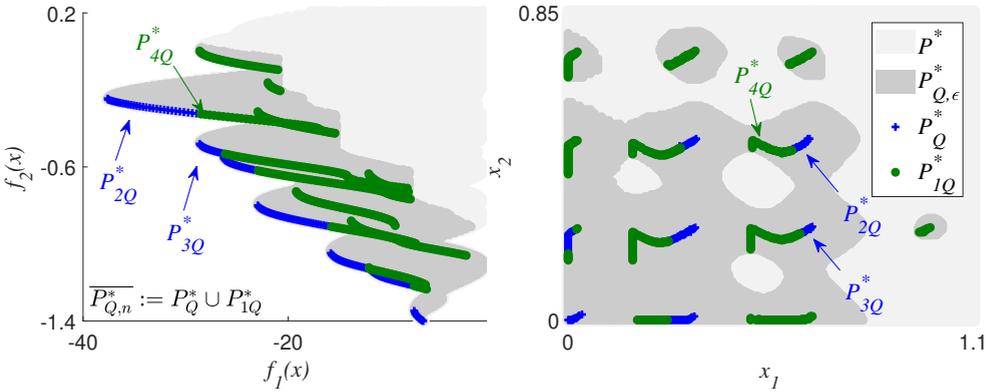
$$f_1(\mathbf{x}) = -\left(\sum_{i=1}^2 10 - 9\cos(2\pi \cdot k_i \cdot x_i)\right)\left(1 - \sqrt{(x_1 - 0.65)^2 + (x_2 - 0.5)^2}\right)$$

$$f_2(\mathbf{x}) = x_1 + x_2 - 1.4,$$

y donde  $k_1 = 2$  y  $k_2 = 3$ , sujeto a:

$$\underline{\mathbf{x}} = [0, 0]$$

$$\overline{\mathbf{x}} = [2, 2]$$

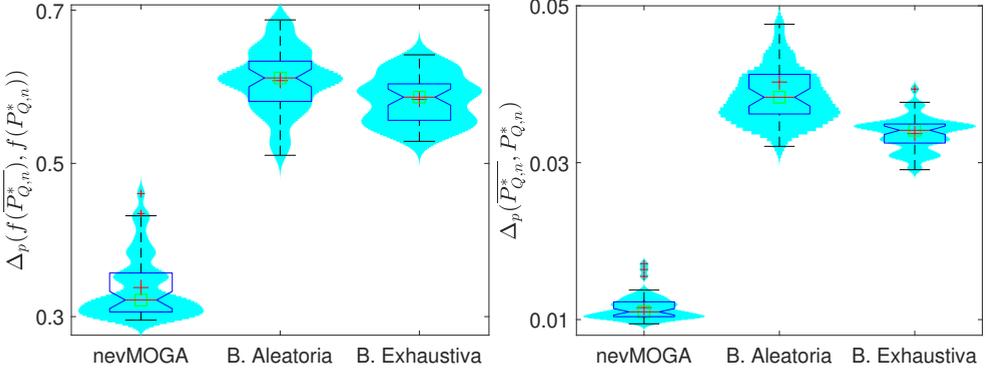


**Figura 3.11:** El conjunto objetivo  $\overline{P_{Q,n}^*}$  está formado por  $P_Q^* \cup P_{1Q}^*$ . El conjunto  $P_{4Q}^*$  está dominado por el conjunto  $P_{3Q}^*$ , sin embargo, se encuentran en vecindarios diferentes y, por este motivo, ambos conjuntos pertenecen al conjunto  $\overline{P_{Q,n}^*}$ .

Primero, y con el objetivo de establecer los dos parámetros de nevMOGA ( $\epsilon$  y  $\mathbf{n}$ ), se realiza una primera búsqueda de soluciones dentro del rango definido. Esto nos lleva a elegir  $\epsilon = [7.7 \ 0.3]$ . Una vez definido  $\epsilon$ , tomamos una solución de referencia  $\mathbf{x}^R = [0.66 \ 0.5]$  y siguiendo el procedimiento detallado en la sección 3.2, obtenemos  $\mathbf{n} = [0.15 \ 0.15]$ . La selección de  $\epsilon$  y  $\mathbf{n}$  lleva a un conjunto objetivo en particular de  $P_{Q,n}$ . En la figura 3.11 se muestra el conjunto objetivo  $\overline{P_{Q,n}^*}$  (discretización precisa de  $P_{Q,n}$ ), necesario para evaluar los algoritmos mediante  $\Delta_p$ . Como se puede ver en dicha figura, el conjunto  $P_{4Q}^*$  está dominado por el conjunto  $P_{3Q}^*$  pero ambos se encuentran ubicados en vecindarios diferentes. El conjunto  $P_{4Q}^*$  no tiene ninguna solución dominada en su vecindad y, por lo tanto, es parte del conjunto objetivo  $\overline{P_{Q,n}^*}$ .

Para el cálculo de cada conjunto  $P_{Q,n}^*$ , obtenido mediante nevMOGA, se ha configurado nevMOGA con los siguientes parámetros:

$$\begin{aligned}
 P_{c/m} &= 0.2 \\
 \mathbf{n\_box} &= [50 \ 50] \\
 N_{ind_G} &= 8 \\
 N_{ind_P} &= 600 \\
 \text{Iteraciones} &= 550
 \end{aligned}$$



**Figura 3.12:** Boxplots para el *benchmark 2*, comparando nevMOGA, búsqueda aleatoria y búsqueda exhaustiva. A la izquierda,  $\Delta_p(f(\overline{P_{Q,n}^*}), f(P_{Q,n}^*))$  mide la aproximación del conjunto obtenido hacia el conjunto objetivo en su imagen en el espacio de objetivos. A la derecha,  $\Delta_p(\overline{P_{Q,n}^*}, P_{Q,n}^*)$  mide la aproximación del conjunto obtenido hacia el conjunto objetivo.

el resto de parámetros toman el valor por defecto sugerido en [37] para el algoritmo original (ev-MOGA).

La figura 3.12, análoga a la figura 3.7, muestra los resultados estadísticos obtenidos por los tres algoritmos comparados sobre el *benchmark 2*. Cada algoritmo se ejecuta 50 veces y evalúa la función objetivo 5000 veces en cada ejecución. Nuevamente, nevMOGA logra mejores resultados que la búsqueda aleatoria y exhaustiva con respecto a la métrica  $\Delta_p$ , ya que alcanza valores más cercanos a cero. Tal y como se realizó para el *benchmark 1*, se ha realizado una prueba de comparación múltiple de estos resultados. Esta comparación muestra que los resultados obtenidos por las tres estrategias de búsqueda son significativamente diferentes en ambos espacios. Por estas razones, se puede concluir que nevMOGA supera significativamente la búsqueda aleatoria y exhaustiva en este *benchmark*.

Para comparar gráficamente los tres algoritmos, se ha elegido un conjunto obtenido  $P_{Q,n}^*$  (de las 50 ejecuciones) de cada uno de ellos. El conjunto  $P_{Q,n}^*$  escogido es el que tiene valores de  $\Delta_p(P_{Q,n}^*, \overline{P_{Q,n}^*})$  y  $\Delta_p(f(P_{Q,n}^*), f(\overline{P_{Q,n}^*}))$  más cercanos a los valores promedio. En las figuras 3.13, 3.14 y 3.15 se observan estos conjuntos comparados con el conjunto objetivo  $\overline{P_{Q,n}^*}$ . En esta figura, se observa claramente que nevMOGA logra una mejor aproximación al conjunto objetivo  $\overline{P_{Q,n}^*}$  (valores más cercanos a 0). Además, el conjunto obtenido por nevMOGA tiene 73 soluciones, mientras que los obtenidos por la búsqueda

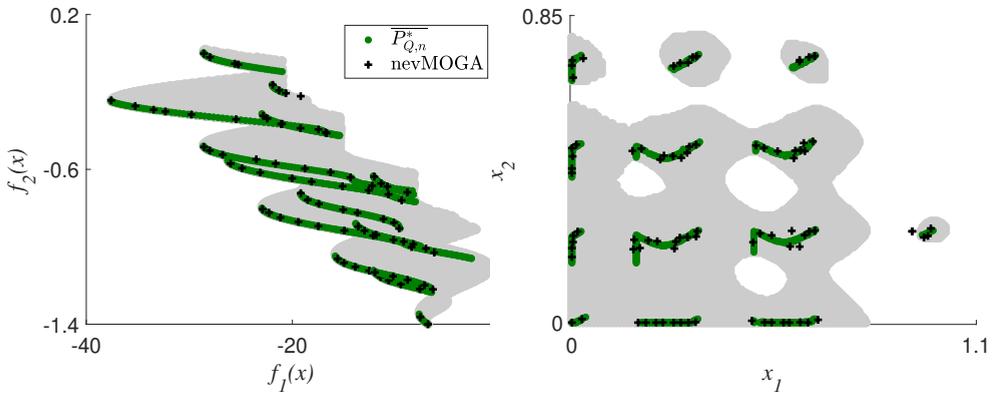
**Tabla 3.2:** Valores numéricos de los resultados estadísticos presentados en la figura 3.12.

	$\Delta_p(f(P_{Q,n}^*), f(\overline{P_{Q,n}^*}))$		
	nevMOGA	Aleatoria	Exhaustiva
Máximo	0.460	0.687	0.641
Q3	0.357	0.633	0.601
Media	0.338	0.609	0.584
Mediana	0.322	0.612	0.584
Q1	0.306	0.581	0.555
Mínimo	0.295	0.511	0.529
	$\Delta_p(P_{Q,n}^*, \overline{P_{Q,n}^*})$		
	nevMOGA	Aleatoria	Exhaustiva
Máximo	0.0322	0.0697	0.0394
Q3	0.012	0.0412	0.0348
Media	0.0119	0.0403	0.0337
Mediana	0.0110	0.0383	0.0340
Q1	0.0104	0.0362	0.0325
Mínimo	0.00946	0.0321	0.0291

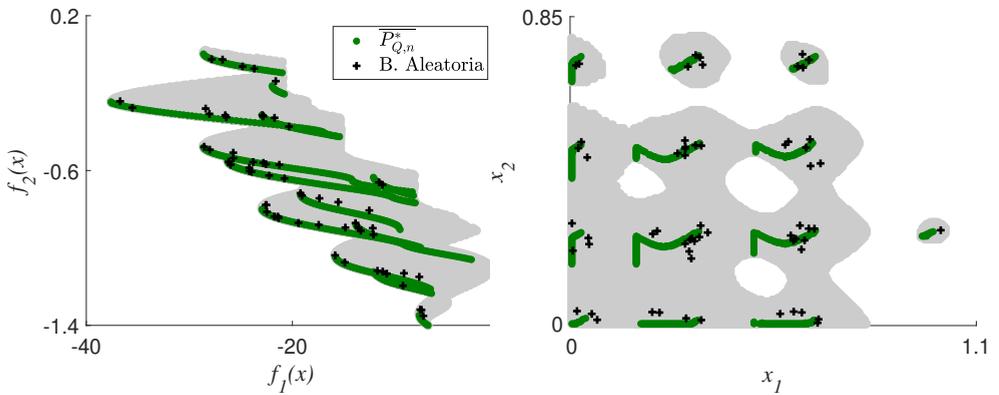
aleatoria y exhaustiva tienen 62 y 48 soluciones respectivamente. Estos dos últimos hechos (una mejor aproximación al conjunto de objetivos  $P_{Q,n}^*$  y un mayor número de soluciones) confirman la conclusión extraída cuando se examinaron estadísticamente los resultados: nevMOGA supera los resultados de sus competidores sobre el *benchmark 2*.

### 3.6 Conclusiones

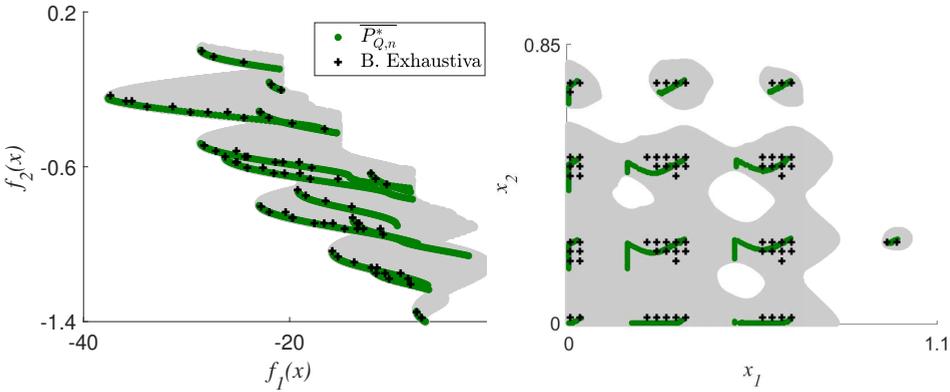
En este capítulo se ha presentado un nuevo algoritmo evolutivo multiobjetivo llamado nevMOGA, que es capaz de obtener el conjunto de solución óptimas y casi-óptimas no dominadas en su vecindad. El diseño de este algoritmo surge de la necesidad de obtener las soluciones que, siendo similares en los objetivos de diseño, son significativamente diferentes en el espacio de parámetros (potencialmente útiles). Estas soluciones pueden ser descuidadas por los algoritmos evolutivos, existentes en la bibliografía, destinados a la localización exclusivamente de alternativas casi-óptimas. Estos algoritmos discretizan solo en función del espacio de objetivos. Mediante este enfoque, las alternativas casi-óptimas significativamente diferentes (en el espacio de parámetros) pueden ser descartadas, ignorando alternativas valiosas para el diseñador que puede condicionar



**Figura 3.13:** Comparación de un conjunto  $P_{Q,n}^*$ , obtenido mediante nevMOGA, y el conjunto objetivo  $\overline{P_{Q,n}^*}$ , en el espacio de parámetros (derecha) y objetivos (izquierda), sobre el *benchmark 2*.



**Figura 3.14:** Comparación de un conjunto  $P_{Q,n}^*$ , obtenido mediante búsqueda aleatoria, y el conjunto objetivo  $\overline{P_{Q,n}^*}$ , en el espacio de parámetros (derecha) y objetivos (izquierda), sobre el *benchmark 2*.



**Figura 3.15:** Comparación de un conjunto  $P_{Q,n}^*$ , obtenido mediante búsqueda exhaustiva, y el conjunto objetivo  $\overline{P_{Q,n}^*}$ , en el espacio de parámetros (derecha) y objetivos (izquierda), sobre el *benchmark* 2.

la decisión final. nevMOGA discretiza en función del espacio de objetivos y parámetros simultáneamente, garantizando que las soluciones potencialmente útiles no son descartadas. Además, nevMOGA descarta las alternativas dominadas en su vecindad, obteniendo un número más manejable de soluciones, lo que permite: un proceso de optimización más eficiente y una etapa de decisión más sencilla.

nevMOGA se ha comparado con otros dos algoritmos (búsqueda aleatoria y búsqueda exhaustiva) sobre dos *benchmarks* planteados, con el objetivo de evaluar su rendimiento. Para realizar esta comparación, se ha empleado la métrica  $\Delta_p$ , que puede medir la calidad de aproximación entre un conjunto obtenido por un algoritmo y el conjunto objetivo  $\overline{P_{Q,n}^*}$  (discretización precisa de  $P_{Q,n}$ ) en el espacio objetivo y parámetros. Además, se ha realizado un análisis estadístico para evitar, en la medida de lo posible, la componente aleatoria de los algoritmos analizados. Los resultados revelan el rendimiento superior de nevMOGA al caracterizar  $P_{Q,n}$  en ambos espacios.

Debido a las diferencias entre  $P_{Q,n}$  (el conjunto objetivo de nevMOGA) y  $P_{Q,\epsilon}$  (el conjunto que caracterizan los algoritmos existentes destinados a la localización de soluciones casi-óptimas) no hubiese sido posible una comparación entre algoritmos que caracterizan dichos conjuntos. Es por este motivo por lo que no se ha llevado a cabo la comparación entre nevMOGA y ningún algoritmo que caracterice  $P_{Q,\epsilon}$ .

## Capítulo 4

# Particularidades del uso de nevMOGA en problemas de optimización multiobjetivo



## 4.1 Introducción

En el capítulo anterior se presentó el algoritmo de optimización multiobjetivo nevMOGA, diseñado para la localización, además del conjunto de Pareto, de las soluciones casi-óptimas no dominadas en su vecindad ( $P_{Q,n}$ ). Las alternativas casi-óptimas pueden ser útiles para el diseñador en cualquier MOP, sin embargo, su utilidad es todavía más relevante ante problemas donde:

1. Existen soluciones multimodales o alternativas casi-óptimas con el mismo valor objetivo.
2. Existen objetivos de diseño formados como la agregación de objetivos parciales.
3. Se excluyen algunos objetivos (debido a decisión del diseñador, para reducir los recursos computacionales o simplificar el análisis de los resultados) de la etapa de optimización que sí se desean contemplar en la fase de toma de decisión.

Además, las soluciones proporcionadas por nevMOGA pueden mostrar al diseñador información relevante que manifieste la existencia de sobreparametrización en el problema definido.

Este capítulo se estructurará de la siguiente forma: en la sección 4.2 se analiza el caso de un MOP con soluciones multimodales o soluciones casi-óptimas con idéntico valor objetivo. Posteriormente, en la sección 4.3, la utilidad del conjunto  $P_{Q,n}^*$  es estudiada cuando se agregan objetivos parciales en los objetivos de diseño. En la sección 4.4 se analiza la utilidad de este conjunto cuando se desea estudiar algún objetivo no incluido en la fase de optimización. Más adelante, en la sección 4.5, se estudia cómo el conjunto  $P_{Q,n}^*$  puede ayudar a detectar la sobreparametrización en un MOP. Por último, se realizan las conclusiones.

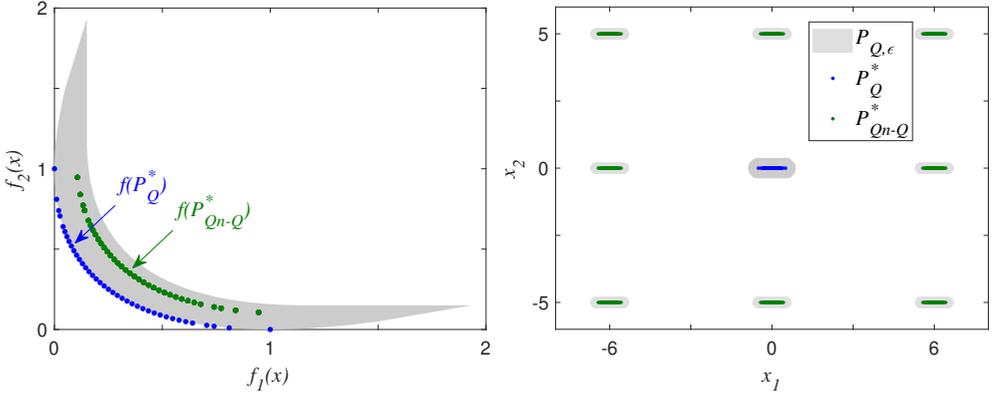
## 4.2 Multimodalidades

En un MOP es posible que existan:

1. Soluciones multimodales (soluciones óptimas con parámetros diferentes que proporcionen el mismo valor objetivo).
2. Soluciones casi-óptimas no dominadas en su vecindad que proporcionan el mismo valor en los objetivos de diseño.

La mayoría de algoritmos de optimización multiobjetivo, ante dos o más soluciones multimodales (o dos o más soluciones casi-óptimas con mismo valor objetivo utilizando algoritmos que obtienen soluciones casi-óptimas) solo obtienen una de ellas. Esto se debe a que la discretización se realiza solo sobre el espacio de objetivos. Sin embargo, las alternativas descartadas pueden ser igual de útiles, que las seleccionadas, para el diseñador. Por ello, el algoritmo empleado no debería descartar ninguna alternativa por tener un valor objetivo idéntico (o muy similar) a otra alternativa, significativamente diferente (no vecina), encontrada previamente. El diseñador debe obtener siempre estas soluciones para poder llevar a cabo la toma de decisión con mayor información. nevMOGA, a diferencia de la mayoría de algoritmos de optimización multiobjetivo, discretiza en función del espacio de objetivos y parámetros simultáneamente. Gracias a ello, nevMOGA no descartará una solución encontrada, del conjunto  $P_{Q,n}$ , por el simple hecho de tener el mismo (o muy similar) valor objetivo que una alternativa no vecina previamente encontrada. Dos o más soluciones vecinas (soluciones con parámetros similares) que tienen idéntico o muy similar valor objetivo, son consideradas soluciones muy similares, y por ello, solo una de ellas será proporcionada por nevMOGA.

En la figura 4.1 se observa el conjunto  $P_{Q,n}^*$  obtenido mediante nevMOGA para el *benchmark* 1 definido en la sección 3.5.2. El conjunto  $P_{Q,n}$  para este problema definido está representado por alternativas en 9 vecindarios diferentes (ver ecuación 3.8). Las alternativas casi-óptimas no dominadas en su vecindad se encuentran representadas en 8 vecindarios diferentes, donde todos ellos obtienen el mismo subconjunto en el espacio de objetivos ( $\mathbf{f}(P_{1Q}^*)$ ). Por lo tanto, existen 8 alternativas que proporcionan el mismo valor objetivo, en cada solución casi-óptima representada en el espacio de objetivos (una por cada uno de los 8 vecindarios). Estos vecindarios podrían ser ignorados si solo se discretiza el espacio de objetivos, ya que no se tiene en cuenta su ubicación en el espacio de parámetros. No parece razonable que el algoritmo empleado decida entre una de estas 8 alternativas sin ningún criterio. Por ello, nevMOGA proporciona todas estas alternativas al diseñador, para que lleve a cabo una decisión más informada. Lo mismo ocurriría con soluciones multimodales (alternativas óptimas con mismo valor objetivo).

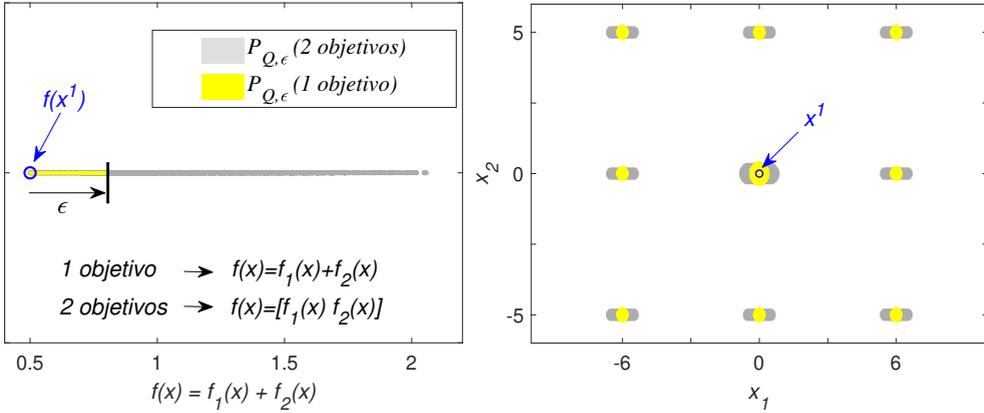


**Figura 4.1:** Conjunto discreto de soluciones  $P_{Q,n}^* = P_Q^* \cup P_{Qn-Q}^*$  obtenidas mediante nevMOGA para el *benchmark* 1. Consta de 9 vecindarios, uno de ellos óptimo  $P_Q^*$ .

### 4.3 Agregación de objetivos de diseño

Es común encontrar MOPs con bastantes objetivos a optimizar. Por este motivo, en muchas ocasiones se agregan algunos de ellos con el fin de reducir su número, simplificando el proceso de optimización y la etapa de decisión. El caso más común es agregar todos los objetivos en un único objetivo de diseño, transformando el problema multiobjetivo en un problema con un único objetivo [85]. Cuando se tienen objetivos agregados es común que se generen multimodalidades y soluciones casi-óptimas no vecinas. Este hecho se debe a la aparición de alternativas que presentan el mismo o similar valor objetivo, pero que tienen diferente *trade-off* entre los objetivos agregados. Además, por haber agregado objetivos, muchas soluciones que antes (sin agregación) eran óptimas pasan a ser casi-óptimas. Todas estas soluciones pueden ser potencialmente útiles ya que proporcionan al diseñador alternativas dignas de estudio en la etapa de decisión. Supongamos ahora que planteamos el *benchmark* 1 definido en la sección 3.5.2. El conjunto  $P_{Q,n}$  definido en la ecuación 3.8 es el conjunto de soluciones óptimas y casi-óptimas. Para analizar la problemática que surge al agregar objetivos de diseño, definimos ahora el problema con un único objetivo de diseño  $f(\mathbf{x})$ , donde se agregan los dos objetivos definidos previamente en dicho *benchmark*:

$$\min_{\mathbf{x}} f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x}) \quad (4.1)$$

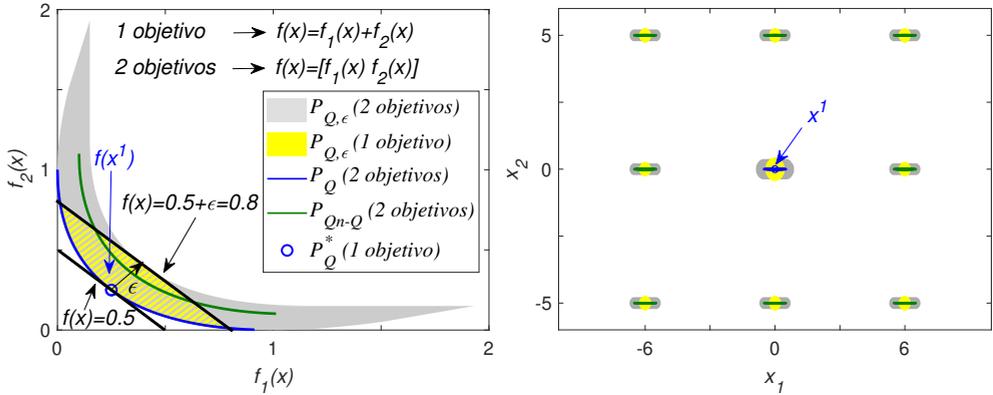


**Figura 4.2:** Benchmark 1 resuelto como un problema con un único objetivo donde  $f(\mathbf{x}) = f_1(\mathbf{x}) + f_2(\mathbf{x})$ .  $\mathbf{x}^1$  es la solución óptima del problema y la zona amarilla es la zona de soluciones casi-óptimas  $P_{Q,\epsilon}$  con el nuevo enfoque. La zona gris corresponde a la proyección de las soluciones casi-óptimas  $P_{Q,\epsilon}$  del MOP original de dos objetivos, sobre el objetivo  $f(\mathbf{x})$ .

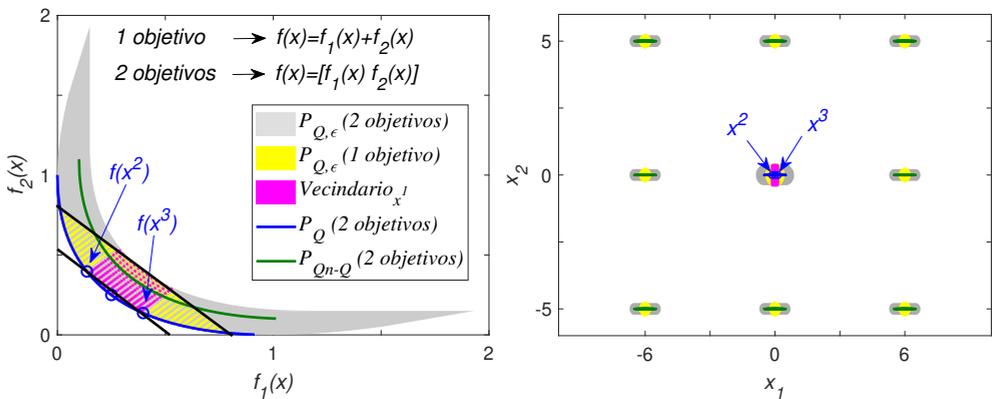
En la figura 4.2 se observa la única solución óptima ( $\mathbf{x}^1$ ) para el problema planteado. Muchas soluciones del frente de Pareto original (con el MOP de dos objetivos) pasan a ser soluciones casi-óptimas en el nuevo escenario. Por lo tanto, si solo se obtiene el conjunto de soluciones óptimas, se perderían muchas alternativas potencialmente útiles.

Sin embargo, al ampliar la zona de interés a  $P_{Q,\epsilon}$  (soluciones casi-óptimas, zona de color amarillo) se puede recuperar gran parte de la información perdida, encontrando las soluciones casi-óptimas no dominadas en su vecindad. En la figura 4.3 se observa la zona de soluciones casi-óptimas  $P_{Q,\epsilon}$  para el escenario de un objetivo sobre el plano  $(f_1, f_2)$ . Esta zona, en el espacio de objetivos, es la zona definida entre las rectas  $f_1(\mathbf{x}) + f_2(\mathbf{x}) = f(\mathbf{x}^1)$  y  $f_1(\mathbf{x}) + f_2(\mathbf{x}) = f(\mathbf{x}^1) + \epsilon$  con  $\epsilon = 0.3$  ( $\epsilon$  elegido para respetar la zona  $P_{Q,\epsilon}$  definida para el MOP original, sección 3.5.2)

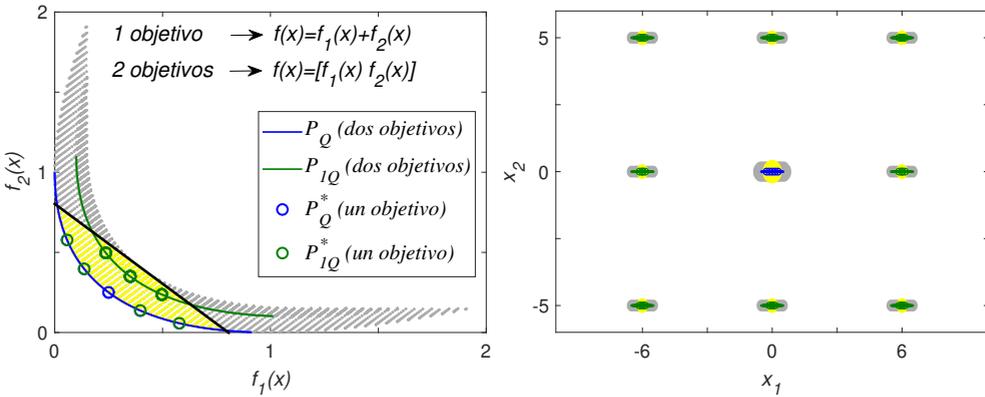
Definiendo la misma vecindad que en el MOP original ( $\mathbf{n} = [0.13 \ 0.38]$ ) se establece la vecindad de  $\mathbf{x}^1$  (ver figura 4.4). Siguiendo nuestro enfoque, todas las soluciones vecinas a  $\mathbf{x}^1$  son soluciones prescindibles (similares a  $\mathbf{x}^1$  pero con peor valor objetivo). Del resto de soluciones casi-óptimas no vecinas a  $\mathbf{x}^1$ , encontramos  $\mathbf{x}^2$  y  $\mathbf{x}^3$  como las soluciones con menor valor  $f(\mathbf{x})$  (ver figura 4.4).



**Figura 4.3:** Comparación del conjunto  $P_{Q,\epsilon}$  definido sobre el problema con un único objetivo ( $f(x) = f_1(x) + f_2(x)$ ) y multiobjetivo ( $f(x) = [f_1(x) \ f_2(x)]$ ).



**Figura 4.4:** Proceso de obtención del conjunto discreto  $P_{Q,n}^*$ .



**Figura 4.5:** Conjunto discreto  $P_{Q,n}^* = P_Q^* \cup P_{Qn-Q}^*$  con  $\epsilon = 0.3$  y  $\mathbf{n} = [0.13 \ 0.38]$  obtenido para el *benchmark* 1 agregando los objetivos de diseño. Pese a la agregación de los objetivos se consigue caracterizar en gran medida el conjunto ideal  $P_{Q,n} = P_Q \cup P_{Qn-Q}$  definido para el problema con dos objetivos.

Siguiendo este procedimiento iterativo se puede caracterizar, en gran medida, el conjunto de interés  $P_{Q,n}$  sin la agregación de objetivos (ecuación 3.8) como se puede observar en la figura 4.5. Esta caracterización puede tener mayor número de soluciones reduciendo la vecindad del problema. Mediante los pesos utilizados para agregar los objetivos se pueden modificar las rectas que definen la zona de soluciones óptimas y casi-óptimas (zona amarilla) hacia una zona diferente del frente, acorde a las preferencias del diseñador.

Por lo tanto, en un MOP al agregar objetivos, se simplifica la etapa de decisión a cambio de perder información útil del MOP original (sin agregación). Sin embargo, gran parte de esta información está contenida en el conjunto de soluciones  $P_{Q,n}$  y puede ser recuperada al caracterizar dicho conjunto.

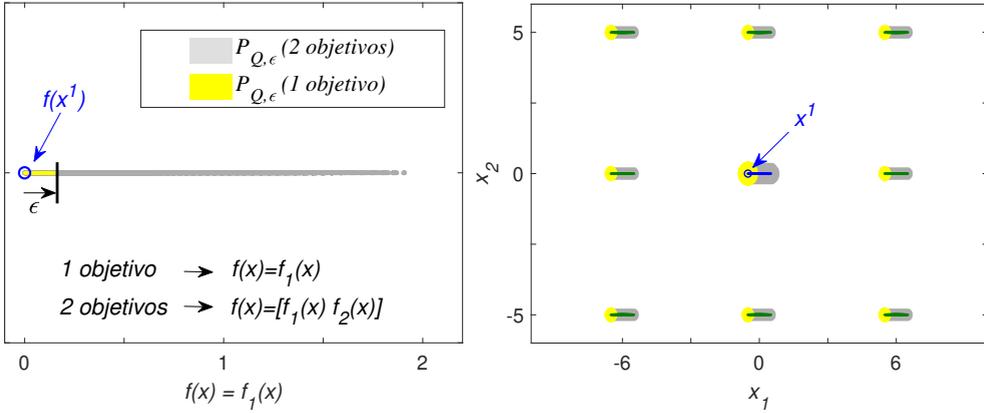
#### 4.4 Exclusión de objetivos de diseño

En un MOP es común estudiar características no incluidas en los objetivos de diseño. Estas características pueden ser estudiadas en la fase de decisión antes de llevar a cabo la decisión final. Estas nuevas características u objetivos no son incluidos en la fase de optimización debido a:

1. Decisión del diseñador: Es común disponer de objetivos que no se desea que influyan en el proceso de optimización. Este es el caso, por ejemplo, de los objetivos relativos a la validación.

2. Limitación de recursos: Hay objetivos que no se pueden incluir en la fase de optimización debido a limitaciones de recursos, de tiempo o económicas. Los objetivos de robustez en sistemas no lineales o de validación sobre una planta real pueden ser un ejemplo. Estos objetivos pueden ser muy costosos computacional o económicamente. En este contexto, se puede eliminar estos objetivos de la fase de optimización e incorporarlos en la etapa de decisión, analizándolos solo sobre el conjunto de soluciones obtenidas en el proceso de optimización. De esta forma se reduce considerablemente el coste computacional o económico.
3. Ayuda a la decisión: Puede ocurrir que un MOP tenga un gran número de objetivos a evaluar. En este contexto es posible que algunos de ellos tengan una importancia secundaria o desconocida. Sin embargo, estos objetivos pueden tomar un papel relevante, en la etapa de decisión, ante dos soluciones con rendimiento muy similar en los objetivos de diseño.

En todos estos contextos, una mejora significativa en los objetivos no incluidos en la fase de optimización puede decantar la elección del diseñador ante dos o más soluciones con rendimiento similar en los objetivos de diseño. Además, el análisis de soluciones casi-óptimas puede informarnos de la buena elección o no de los objetivos de diseño. Analizando las características finales de las soluciones, es posible que soluciones casi-óptimas presenten propiedades más deseadas que las óptimas. Por ejemplo, en el ajuste de controladores la respuesta temporal obtenida por un controlador casi-óptimo puede ser preferida ante la de un controlador óptimo. Esta situación indica la mala elección de los objetivos de diseño, pudiendo llevar al diseñador a replantearse los mismos. También existe la posibilidad de que, por cualquier motivo, se modifique el modelo de optimización ligeramente, alterando así los objetivos y seguramente las soluciones óptimas. En un MOP tradicional, donde se caracteriza exclusivamente el frente de Pareto, solo podríamos analizar los objetivos excluidos (o los modificados) sobre las soluciones óptimas encontradas. Considerar las soluciones casi-óptimas no dominadas en su vecindad añadiría alternativas que presumiblemente tendrán un comportamiento diferente (al ser no vecinas de las que le dominan) respecto a los objetivos no incluidos en la fase de optimización. Por lo tanto, es clave considerar estas soluciones ya que proporcionan información muy valiosa para el estudio de estos objetivos en la fase de decisión. Para ello, primero se calcula el conjunto  $P_{Q,n}^*$  para los objetivos a optimizar. Posteriormente, se analiza el comportamiento de este conjunto sobre los objetivos no incluidos en la etapa de optimización, en la fase de decisión. Este proceso implica una preferencia de los objetivos que intervienen en el proceso de optimización sobre el resto, al no ser considerados para el cálculo del conjunto  $P_{Q,n}^*$ .



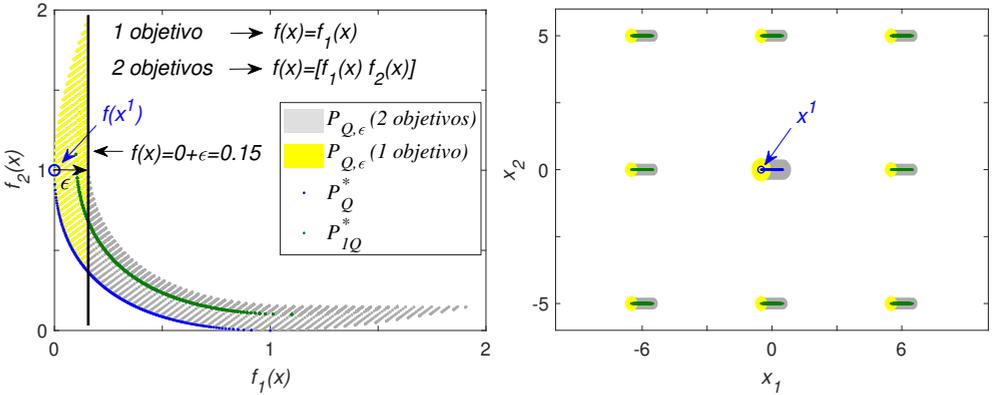
**Figura 4.6:** *Benchmark 1* resuelto como un problema con un único objetivo donde  $f(\mathbf{x}) = f_1(\mathbf{x})$ .  $\mathbf{x}^1$  es la solución óptima del problema y la zona amarilla es la zona de soluciones casi-óptimas  $P_{Q,\epsilon}$  con el nuevo enfoque. La zona gris corresponde a la proyección de las soluciones casi-óptimas  $P_{Q,\epsilon}$  del MOP original de dos objetivos, sobre el objetivo  $f(\mathbf{x})$ .

Supongamos que tenemos de nuevo el *benchmark 1* (sección 3.5.2). Se escoge un único objetivo de diseño  $f(\mathbf{x}) = f_1(\mathbf{x})$ . El objetivo  $f_2(\mathbf{x})$  no interviene en el proceso de optimización y solo se contemplará su aplicación tras la optimización, utilizando la información que se genere en el posterior proceso de toma de decisión:

$$\min_{\mathbf{x}} f(\mathbf{x}) = f_1(\mathbf{x}) \tag{4.2}$$

En la figura 4.6 se observa la visualización de este escenario, donde solo existe una única solución óptima  $\mathbf{x}^1$ . Por lo tanto, si solo obtuviéramos la solución óptima perderíamos mucha información (frente de Pareto y 8 vecindarios del MOP original). Todas estas soluciones podrían ser útiles para el estudio del objetivo no incluido en fase de optimización.

Sin embargo, al ampliar la zona de interés a  $P_{Q,\epsilon}$  (soluciones casi-óptimas) se mantienen parte de esas soluciones de interés. En la figura 4.7 se observa  $P_{Q,\epsilon}$  sobre el problema con un único objetivo (zona de color amarillo). Este conjunto, en el espacio de objetivos, se encuentra delimitado entre la solución óptima  $f(\mathbf{x}^1)$  y la recta  $f_1(\mathbf{x}) = f(\mathbf{x}^1) + \epsilon$  con  $\epsilon = 0.15$  ( $\epsilon$  elegido para respetar la zona  $P_{Q,\epsilon}$  definida para el problema multiobjetivo, sección 3.5.2).



**Figura 4.7:** Comparación de los conjunto  $P_{Q,\epsilon}$  sobre el problema con un único objetivo ( $f(\mathbf{x}) = f_1(\mathbf{x})$ ) y multiobjetivo ( $f(\mathbf{x}) = [f_1(\mathbf{x}) f_2(\mathbf{x})]$ ).

Definiendo la misma vecindad que en el problema multiobjetivo ( $\mathbf{n} = [0.13 \ 0.38]$ ) se establece la vecindad de  $\mathbf{x}^1$  (zona magenta en la figura 4.8) como la zona de soluciones similares a  $\mathbf{x}^1$ . Todas estas soluciones, siendo similares a  $\mathbf{x}^1$  en sus parámetros, tienen peor valor objetivo  $f(\mathbf{x})$  y por lo tanto son descartadas. Una vez descartadas las soluciones pertenecientes a la vecindad de  $\mathbf{x}^1$ , se escogen  $\mathbf{x}^2$  y  $\mathbf{x}^3$  como las soluciones con mejor prestación del resto de alternativas (ver figura 4.8).

De esta forma iterativa se puede caracterizar, en gran medida, el conjunto de interés  $P_{Q,n}$  (problema multiobjetivo) como se puede observar en la figura 4.9. Esta caracterización puede tener mayor número de soluciones reduciendo la vecindad del problema. Además, también sería posible localizar alternativas que no se encuentran en la zona de soluciones casi-óptimas para el problema con un único objetivo (zona de color amarillo) ampliando  $\epsilon$ .

Lógicamente existe una preferencia del objetivo optimizado ( $f_1$ ), sobre el objetivo no incluido en la fase de optimización ( $f_2$ ), al no considerar  $f_2$  para el cálculo del conjunto  $P_{Q,n}^*$ . Sin embargo, se observa como hay soluciones, por ejemplo  $\mathbf{x}^4$  y  $\mathbf{x}^5$ , que a pesar de tener peor prestación sobre el objetivo optimizado, son mejores que  $\mathbf{x}^1$  con respecto al objetivo no incluido en la fase de optimización  $f_2$ . De hecho,  $\mathbf{x}^4$  consigue una mejora significativa sobre  $f_2$  a cambio de una ligera pérdida sobre  $f_1$ . Por lo tanto, estas soluciones ( $\mathbf{x}^4$  y  $\mathbf{x}^5$ ) son dignas de estudio, pues tienen un comportamiento casi-óptimo en el objetivo optimizado y consiguen mejor rendimiento en el objetivo no optimizado  $f_2(\mathbf{x})$ .

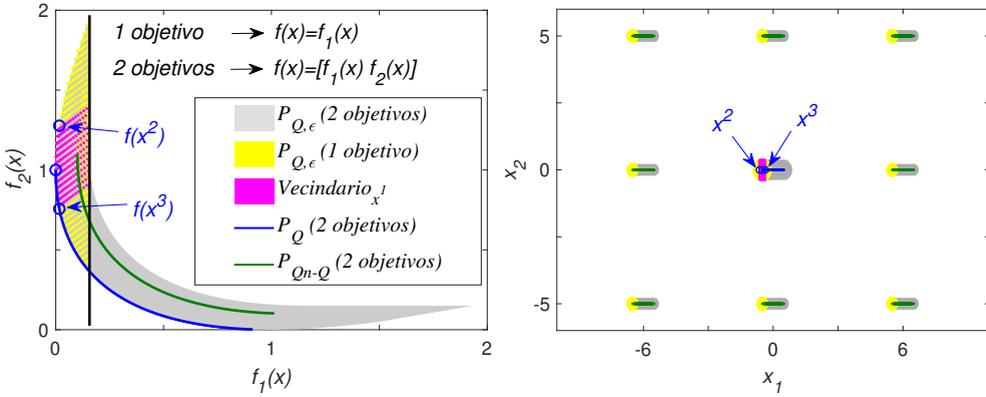


Figura 4.8: Proceso de obtención del conjunto discreto  $P_{Q,n}^*$ .

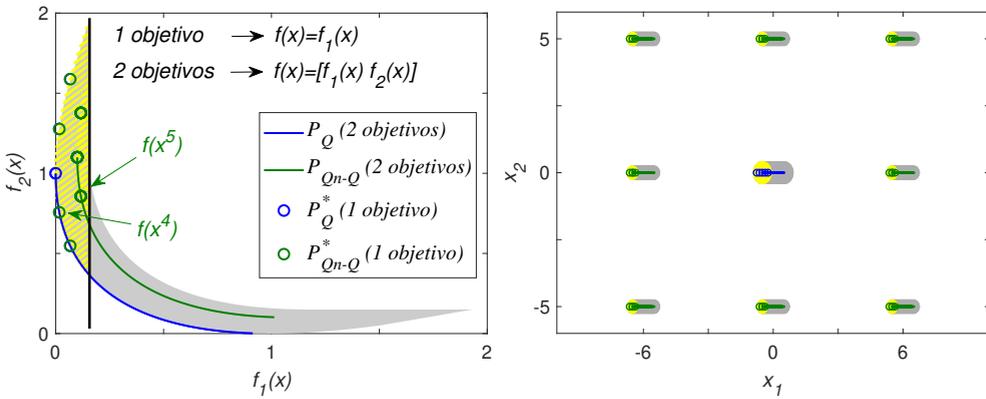


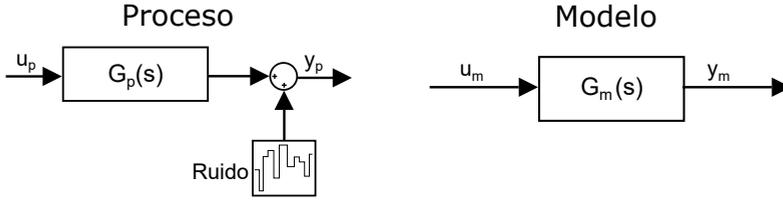
Figura 4.9: Conjunto discreto  $P_{Q,n}^* = P_Q^* \cup P_{Qn-Q}^*$  con  $\epsilon = 0.15$  y  $\mathbf{n} = [0.13 \ 0.38]$ . Pese a la exclusión de la fase de optimización de  $f_2$  se consigue caracterizar en gran medida el conjunto ideal  $P_{Q,n} = P_Q \cup P_{Qn-Q}$  definido para el problema con dos objetivos.

## 4.5 Sobreparametrización

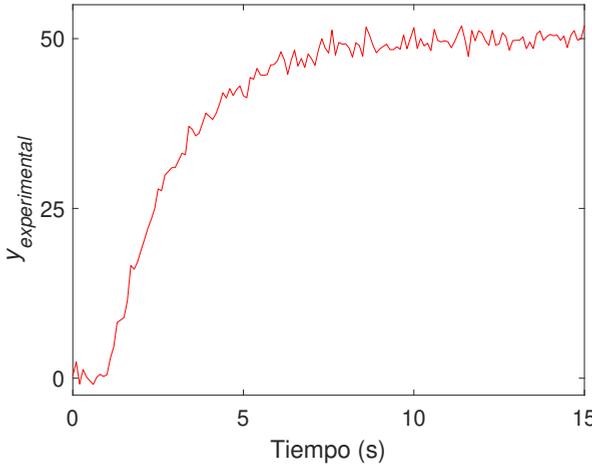
Cuando los MOP son complejos, pueden tener asociados un gran número de variables de decisión. Por ejemplo, en el caso de identificación de modelos que se plantean como MOP, puede ocurrir que los modelos tengan una gran cantidad de parámetros a estimar. Especialmente en estos casos, aunque también en otros no tan complejos, cabe la posibilidad (por una incorrecta definición del problema) de que exista sobreparametrización. Esto ocurre cuando el modelo se define con más parámetros de los necesarios, debido a dependencias entre ellos que no son detectadas por el diseñador. Obteniendo el conjunto discreto de soluciones  $P_{Q,n}^*$  es posible detectar síntomas de este problema. Se va a mostrar esta situación mediante un ejemplo sencillo en el que la sobreparametrización es trivial. En primer lugar se dispone del proceso mostrado a la izquierda de la figura 4.10. Este proceso es un sistema de primer orden con una ganancia  $K_p = 10$ , una constante de tiempo  $\tau_p = 2$  y una señal de ruido agregada a la salida del sistema. Se desea identificar este proceso con el modelo de primer orden que se observa a la derecha de la figura 4.10 y los datos obtenidos del proceso mediante simulación con entrada escalón. Se provoca la sobreparametrización en la ganancia del modelo  $K = K1 \cdot K2$ . De este modo, diferentes combinaciones de estos parámetros ( $K1$  y  $K2$ ) dan la misma ganancia del modelo  $K$ , y por lo tanto son alternativas idénticas. Así, se tienen tres parámetros para identificar el modelo ( $\mathbf{x} = [K1 \ \tau \ K2]$ ).

Esta sobreparametrización puede aparecer, por ejemplo, en identificación de modelos mediante primeros principios. En este contexto, es habitual que determinados coeficientes sean dependientes en las ecuaciones del modelo. Esto ocurre en el modelo de un sistema de refrigeración para el sistema  $\mu$ -CHP (ver Apéndice A) realizado en esta tesis. En este modelo, se tiene la contante  $K_a$ , que es el producto del calor específico ( $c_a$ ) y la densidad del aire ( $\rho_a$ ) dentro de la pila. Estas dos variables no aparecen en ninguna otra ecuación del modelo. Tratar  $c_a$  y  $\rho_a$  como parámetros independientes en el proceso de optimización provocaría una sobreparametrización (desde el punto de vista de la optimización), provocando que infinitas combinaciones de  $c_a$  y  $\rho_a$  den el mismo valor de la contante  $K_a$ . Por lo tanto, estos coeficientes dependientes deben ser introducidos como uno solo y no como coeficientes independientes, para evitar definir un MOP con sobreparametrización.

$$\begin{aligned} G_p(s) &= \frac{K_p}{\tau_p s + 1} = \frac{10}{2s + 1} \\ G_m(s) &= \frac{K}{\tau s + 1} = \frac{K1 \cdot K2}{\tau s + 1} \end{aligned} \quad (4.3)$$



**Figura 4.10:** En la izquierda, proceso de primer orden a identificar. En la derecha, modelo de primer orden utilizado para la identificación.

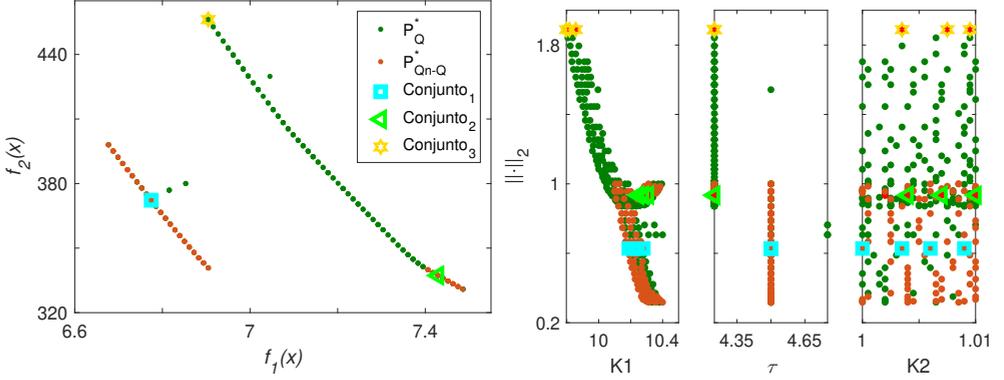


**Figura 4.11:** Datos del ensayo experimental sobre el proceso con  $K_p = 10$ ,  $\tau_p = 2$  y una señal de ruido agregada a la salida del sistema. Se ha introducido un escalón de 5 unidades.

Para el ejemplo de identificación introducido, se dispone de un ensayo experimental (ver figura 4.11) en el que se introduce un escalón de 5 unidades en  $u_p$  para obtener  $y_{experimental}$ . Para identificar los parámetros del modelo con los datos experimentales ( $y_{experimental}$ ) se plantea el siguiente el MOP:

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x})] \tag{4.4}$$

$$\begin{aligned} f_1(\mathbf{x}) &= \max |e(t)| \\ f_2(\mathbf{x}) &= \int_0^t |e(t)| \end{aligned} \tag{4.5}$$



**Figura 4.12:** Conjunto de soluciones  $P_{Q,n}^* = P_Q^* \cup P_{Qn-Q}^*$  con  $\epsilon = [0.4 \ 20]$  y  $\mathbf{n} = [0.1 \ 0.2 \ 0.002]$ . A la derecha, se observa la visualización de las variables de decisión mediante la herramienta de visualización Level Diagrams (LD [9]), utilizando la norma 2. Se observa como conjuntos de soluciones (Conjunto<sub>1</sub>, Conjunto<sub>2</sub> y Conjunto<sub>3</sub>) tienen diferentes soluciones con idéntico valor objetivo.

donde:

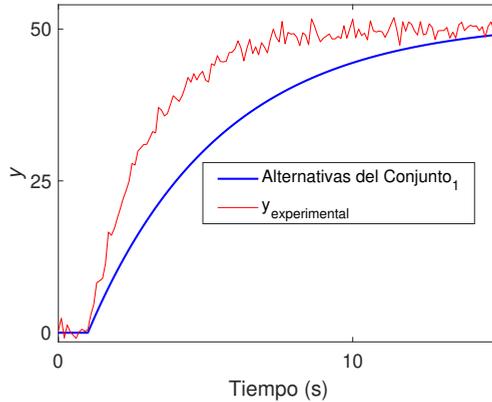
$$\begin{aligned} e(t) &= y_{\text{modelo}}(t) - y_{\text{experimental}}(t) \\ t &= 15 \text{ segundos} \\ \mathbf{x} &= [K1 \ \tau \ K2] \end{aligned}$$

sujeto a:

$$\begin{aligned} \underline{\mathbf{x}} &= [9.6 \ 1 \ 0.99] \\ \bar{\mathbf{x}} &= [10.4 \ 10 \ 1.01] \end{aligned}$$

En el MOP se propone una dependencia en la ganancia del sistema  $K = K1 \cdot K2$ , produciendo una sobreparametrización. El objetivo es encontrar el conjunto de modelos óptimos que consiga minimizar los objetivos propuestos (errores sobre un ensayo experimental) ante un escalón de 5 en  $u$ .

En la figura 4.12 se observa el conjunto discreto  $P_{Q,n}^* = P_Q^* \cup P_{Qn-Q}^*$  obtenido al realizar una búsqueda exhaustiva en el rango de soluciones posibles ( $\underline{\mathbf{x}}$  y  $\bar{\mathbf{x}}$ ). Se observa cómo diferentes soluciones dan el mismo valor para ambos objetivos (Conjunto<sub>1</sub>, Conjunto<sub>2</sub> y Conjunto<sub>3</sub>). Si este fenómeno (soluciones multimodales) se repite sistemáticamente (como ocurre en este caso) puede hacernos



**Figura 4.13:** Respuesta temporal de las alternativas del Conjunto<sub>1</sub>.

pensar que puede haber sobreparametrización. Es posible aumentar el número de objetivos a evaluar. Si persiste este fenómeno (diversas soluciones con mismo valor objetivo) la sobreparametrización es más probable. Además, también es posible observar la respuesta temporal de las alternativas con idéntico valor objetivo. Si todas obtienen la misma respuesta temporal hay más indicios sobre la existencia de sobreparametrización. Por ejemplo, en la figura 4.13 se observa la respuesta temporal de las alternativas del Conjunto<sub>1</sub>, observando que todas ellas tienen la misma respuesta temporal, lo que aporta un nuevo indicio de la posible presencia de sobreparametrización. De esta forma, nevMOGA proporciona información al diseñador acerca de la existencia de esta problemática, que posteriormente deberá ser confirmada mediante un análisis profundo del problema definido.

## 4.6 Conclusiones

En este capítulo se han mostrado casos particulares en los que se pone de manifiesto la ventaja de encontrar el conjunto de soluciones  $P_{Q,n}$ . Estos casos serían los siguientes:

1. MOPs con soluciones multimodales o casi-óptimas con mismo valor objetivo.
2. MOPs con agregación de objetivos.

3. MOPs con objetivos no incluidos en la fase de optimización pero si en la de decisión.
4. MOPs donde existan posibles problemas de sobreparametrización.

Las soluciones óptimas o casi-óptimas con mismo (o muy similar) valor objetivo son soluciones potencialmente útiles y en muchos casos son ignoradas. nevMOGA siempre las considera si se encuentran en vecindarios diferentes.

En los problemas donde se agregan objetivos es común tener una pérdida de información al realizar la agregación (especificaciones a priori mediante pesos). Esta pérdida de soluciones interesantes es en gran parte subsanada al encontrar el conjunto de soluciones  $P_{Q,n}^*$ . Además es posible caracterizar una zona del conjunto de soluciones  $P_{Q,n}$  sin la agregación. Esta zona puede modificarse, variando los pesos utilizados al agregar los objetivos, acorde a las preferencias del diseñador.

Cuando se eliminan objetivos de la fase de optimización o simplemente se tienen objetivos diferentes a estudiar en la fase de decisión, obtener las soluciones  $P_{Q,n}^*$  para los objetivos optimizados permite analizar estos objetivos en soluciones potencialmente útiles. El conjunto  $P_{Q,n}^*$  aporta, presumiblemente, nuevos comportamientos respecto a los objetivos a estudiar en la fase de decisión.

La sobreparametrización es un problema muchas veces difícil de detectar cuando se tienen diversas variables de decisión. Gracias a la obtención del conjunto  $P_{Q,n}^*$  se puede detectar la existencia sistemáticamente de diversas alternativas con mismo valor objetivo (soluciones multimodales). Esto puede hacernos sospechar de la existencia de sobreparametrización. La confirmación de esta problemática se obtiene realizando un análisis profundo sobre el sistema en cuestión, quedando reflejada la dependencia entre dos o más parámetros del sistema.

Por lo tanto, se ha mostrado la utilidad del conjunto  $P_{Q,n}$  en diferentes casos particulares que pueden darse de forma habitual en los MOPs que se nos planteen.



## Capítulo 5

# Aplicación de nevMOGA a problemas en ingeniería de control



## 5.1 Introducción

Una vez introducida la metodología de optimización multiobjetivo presentada en esta tesis, en este capítulo se muestra su aplicación en problemas del ámbito de la ingeniería de control de procesos. En dicho ámbito es común la obtención de un modelo matemático que describa el comportamiento dinámico de un determinado proceso o sistema. Habitualmente se ajustan los parámetros del modelo utilizando datos experimentales del proceso, lo que se conoce como identificación. En control, disponer de un modelo que refleje adecuadamente el comportamiento del sistema es esencial para la calidad del control del sistema. Por ello, en este capítulo se aplicará la metodología introducida en esta tesis sobre problemas de identificación de modelos y ajuste de controladores en sistemas multivariantes lineales y no lineales.

Las tareas de identificación y control guardan ciertas similitudes. La más relevante, desde el punto de vista que nos atañe, es que ambas pueden y suelen plantearse como problemas de optimización. En ambas tareas, dichos problemas de optimización conllevan diferentes objetivos que pueden entrar en conflicto. Por un lado, en el proceso de identificación se suele medir el error entre la salida real (ante un experimento dado) y simulada. Este error puede medirse teniendo en cuenta el valor: máximo, medio, cuartil, etc. Por otro lado, en el ajuste de controladores existe mayor variación de objetivos [33] y suelen estar relacionados con: las prestaciones (error ante cambios de referencia, perturbaciones, etc), la robustez, el esfuerzo de control, etc. En ambas tareas, ante procesos multivariantes los objetivos aumentan rápidamente.

Por lo tanto, en ambas tareas nos encontramos ante MOPs. Los objetivos que se plantean en dichos MOPs pueden estar formados por objetivos agregados. Además, puede existir incertidumbre en los mismos. También es posible disponer de objetivos que no deseamos incluir en la fase de optimización por diversos motivos.

Analicemos esto en más detalle, en un primer lugar, en muchos MOP se agregan objetivos directa o indirectamente. Un ejemplo de ello se produce cuando los modelos o controles son multivariantes, donde el número de objetivos crece rápidamente (se suele agregar los errores en las diferentes salidas, cuando identificamos un modelo y agregar los esfuerzos de control de las entradas del proceso, en tareas de diseño de controladores). Estos objetivos se agregan habitualmente con el fin de simplificar la etapa de optimización y la fase de decisión. Además, esta agregación también puede ser indirecta. Este es el caso, por ejemplo, de los objetivos que se evalúan en el contexto de un experimento o escenario, especialmente cuando trabajamos con sistemas no lineales, que

contempla diferentes situaciones (diferentes puntos de operación, etc). En este contexto, la obtención del conjunto  $P_{Q,n}$  puede ser relevante, ya que proporciona al diseñador alternativas con rendimiento similar, pero con un comportamiento significativamente diferente (diferente compensación entre los objetivos agregados).

En segundo lugar, es posible que los objetivos de diseño tengan incertidumbre. Cuando esto ocurre, un simple cambio en la configuración de determinados parámetros (por ejemplo, en el entorno simulación) puede variar el frente de Pareto. En este escenario, soluciones casi-óptimas pueden ser óptimas con mínimas variaciones de estos parámetros. Por ello, cuando existe incertidumbre, las alternativas casi-óptimas pueden ser también relevantes.

Por último, estudiar en la fase de decisión objetivos no incluidos en la fase de optimización es una práctica común. Un diseñador puede decidir excluir un objetivo de la fase de optimización por diferentes motivos (importancia secundaria, desconocimiento a priori, validación, etc). Además, el estudio de objetivos no incluidos en la fase de optimización puede deberse a limitaciones económicas (por ejemplo, objetivos medidos sobre una planta real) o temporales (por ejemplo, objetivos computacionalmente costosos). En todos estos casos, el conjunto  $P_{Q,n}$  puede aportar soluciones similares en los objetivos de diseño y con rendimiento significativamente mejor en los objetivos no incluidos en la fase de optimización.

Este capítulo se estructura de la siguiente forma: en la sección 5.2, se muestra un ejemplo de diseño de controladores PI para un sistema multivariable. En la sección 5.3, se diseña el control para la planta de destilación Wood & Berry. Posteriormente, en la sección 5.4, se diseña un control para el sistema de refrigeración que se describe en el concurso de ingeniería CIC2018. A continuación, en la sección 5.5, se realiza la identificación del sistema no lineal de refrigeración de un sistema  $\mu$ -CHP basado en una pila PEMFC. En la sección 5.6, se diseña el control de las temperaturas del agua de entrada y salida de la pila PEMFC para el modelo identificado anteriormente. Por último, se presentan las conclusiones.

## 5.2 Ejemplo de diseño de un controlador multivariable

En esta sección se plantea un sistema de control multivariable para analizar la utilidad de la metodología planteada en la tesis. En primer lugar se definirá el MOP. Posteriormente, mediante nevMOGA, se obtendrán el conjunto de controladores óptimos y casi-óptimos no dominados en su vecindad. Finalmente se analizarán los controladores obtenidos, examinando especialmente los controladores casi-óptimos (ignorados en un MOP tradicional).

El problema de diseño planteado consiste en seleccionar dos controladores PI, con el objetivo de optimizar indicadores de prestaciones y esfuerzo de control sobre una planta multivariable con dos entradas y dos salidas. El modelo de la planta es [57]:

$$Y(s) = \begin{pmatrix} y_1(s) \\ y_2(s) \end{pmatrix} = G(s) \begin{pmatrix} u_1(s) \\ u_2(s) \end{pmatrix} \quad (5.1)$$

$$G(s) = \begin{pmatrix} \frac{5e^{-40}}{100s+1} & \frac{1e^{-4}}{10s+1} \\ \frac{-5e^{-40}}{10s+1} & \frac{5e^{-40}}{100s+1} \end{pmatrix}$$

donde  $y_1$  y  $y_2$  son las salidas de la planta, y  $u_1$  y  $u_2$  son las entradas (acciones de control). Las constantes de tiempo están en segundos.

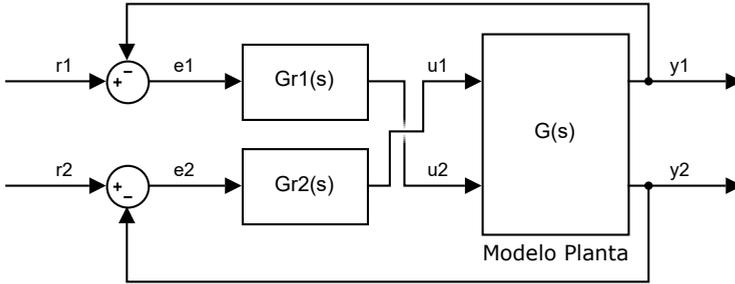
La estructura de control (también utilizada en [57]) se define como un emparejamiento *off-diagonal* de controladores PI, es decir, la salida  $y_1$  se controla mediante la acción de control  $u_2$  e  $y_2$  mediante  $u_1$ . Esto es:

$$\begin{aligned} u_2(s) &= K_{c1} \left( e_1(s) + \frac{1}{T_{i1}} \frac{1}{s} e_1(s) \right) \\ u_1(s) &= K_{c2} \left( e_2(s) + \frac{1}{T_{i2}} \frac{1}{s} e_2(s) \right) \end{aligned} \quad (5.2)$$

donde  $K_{c1}$  y  $K_{c2}$  son las ganancias proporcionales,  $T_{i1}$  y  $T_{i2}$  son las constantes de tiempo integrales en segundos y  $e_1 = r_1 - y_1$  y  $e_2 = r_2 - y_2$  son los errores, donde  $r_1$  y  $r_2$  son las referencias para  $y_1$  y  $y_2$  respectivamente (ver figura 5.1).

Como se ha comentado anteriormente, agregando los objetivos de diseño conseguimos simplificar los procesos de optimización y decisión. Por ello, ante problemas con diversos objetivos, parece lógico establecer el MOP con objetivos de diseño formados por objetivos agregados. Se define el MOP como:

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x})] \quad (5.3)$$



**Figura 5.1:** Esquema de emparejamiento de controladores PI (*off-diagonal*) empleado para el control del sistema.

donde

$$\begin{aligned}
 f_1 &= \int_0^{t_f} e_1^2 + e_2^2 \Big|_{r_1=1, r_2=0} dt + \int_0^{t_f} e_1^2 + e_2^2 \Big|_{r_1=0, r_2=1} dt \\
 f_2 &= \int_0^{t_f} u_1^2 + u_2^2 \Big|_{r_1=1, r_2=0} dt + \int_0^{t_f} u_1^2 + u_2^2 \Big|_{r_1=0, r_2=1} dt
 \end{aligned} \tag{5.4}$$

sujeto a:

$$\begin{aligned}
 \underline{\mathbf{x}} &\leq \mathbf{x} \leq \bar{\mathbf{x}} \\
 f_1(\mathbf{x}) &< 300, f_2(\mathbf{x}) < 200 \\
 &Estable en bucle cerrado
 \end{aligned}$$

y donde

$$\begin{aligned}
 \mathbf{x} &= [Kc_1, Ti_1, Kc_2, Ti_2] \\
 \underline{\mathbf{x}} &= [-1, 1, 0.1, 1] \quad \bar{\mathbf{x}} = [-0.1, 200, 10, 1000] \\
 t_f &= 1000 \text{ segundos}
 \end{aligned}$$

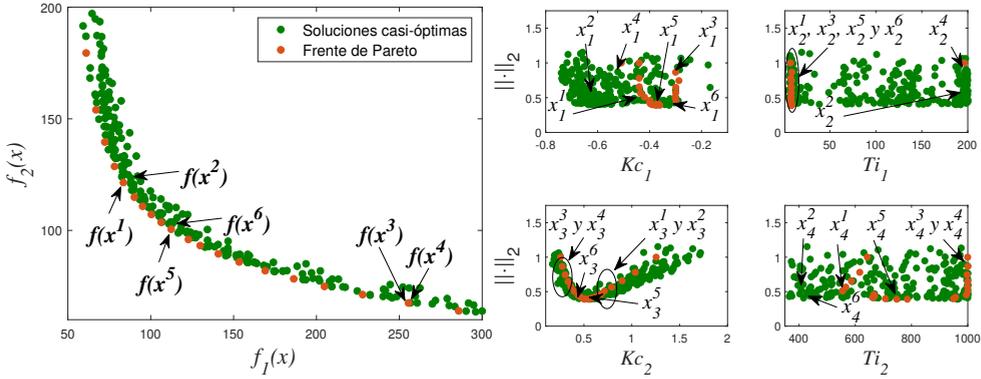
Se realizan dos simulaciones diferentes, secuencialmente. En la primera, se aplica un escalón unitario a  $r_1$  mientras que  $r_2$  se mantiene en cero. En la segunda, el escalón se aplica a  $r_2$ , manteniendo  $r_1$  a cero. En cada simulación, la integral del cuadrado del error (ISE) se calcula para las salidas  $y_1$  e  $y_2$ .

Nuestro primer objetivo  $f_1$  se define como la agregación de los valores del ISE de cada salida sobre cada simulación (4 términos de  $f_1$ ). Por lo tanto, el objetivo  $f_1$  mide el seguimiento de referencia del sistema de control. El segundo objetivo  $f_2$  se calcula de manera similar, utilizando las acciones de control en lugar de los errores obtenidos. Por lo tanto,  $f_2$  mide el esfuerzo de control en cada acción de control sobre cada simulación (4 términos de  $f_2$ ). Por lo tanto, cada objetivo ( $f_1$  y  $f_2$ ) se puede ver como una agregación de 4 objetivos (ver ecuación 5.4).

Una vez definido el problema de optimización, se deben establecer los dos parámetros principales de nevMOGA ( $\epsilon$  y  $\mathbf{n}$ ). Es importante elegir estos parámetros correctamente, especialmente el parámetro  $\mathbf{n}$ . En primer lugar, tenemos que decidir qué pérdida de rendimiento (con respecto al conjunto de Pareto) estamos dispuestos a aceptar (es decir, el parámetro  $\epsilon$ ). En otras palabras, hasta donde vamos a considerar una solución como casi-óptima. La elección de un valor excesivamente alto llevaría a la obtención de un número muy elevado de soluciones, mientras que un valor excesivamente bajo proporcionaría muy pocas. En este ejemplo de aplicación, aunque los objetivos de diseño tienen un sentido físico, no es evidente elegir  $\epsilon$ . Para resolver esta dificultad hemos procedido de la siguiente manera. Primero, hemos tomado los valores de  $f_1$  y  $f_2$  correspondientes al control diseñado en [57] como una solución de referencia. Este controlador de referencia es el que minimiza la suma de los objetivos de diseño ( $f_1 + f_2$ ). Luego, hemos identificado esa solución en el frente de Pareto (tomado de [40]). Finalmente, sobre la base de esa información, hemos decidido elegir  $\epsilon = [10 \ 5]$ . Tenga en cuenta que hay un elemento subjetivo inevitable sobre la forma en que se elige  $\epsilon$ . Ahora, tenemos que asignar un valor al vector  $\mathbf{n}$ , que son los valores a partir de los cuales dos controladores se consideran significativamente diferentes. Por un lado, la elección de un valor excesivamente bajo haría que la mayoría de las soluciones no fueran soluciones vecinas, lo que a su vez provocaría que el algoritmo tendiera a obtener el conjunto completo  $P_{Q,\epsilon}$ . Por otro lado, elegir un valor excesivamente alto daría como resultado que muchas soluciones sean soluciones vecinas, lo que provocaría que el algoritmo tienda a obtener solo el conjunto óptimo  $P_Q$ . Es por esto que es crucial que  $\mathbf{n}$  sea elegido cuidadosamente, apoyando la elección con buenas razones. Siguiendo el procedimiento detallado en la sección 3.2 obtenemos  $\mathbf{n} = [0.25 \ 50 \ 0.1 \ 200]$ .

La configuración del resto de parámetros de nevMOGA es la siguiente:

- $Nind_G = 32$
- $Nind_P = 100$



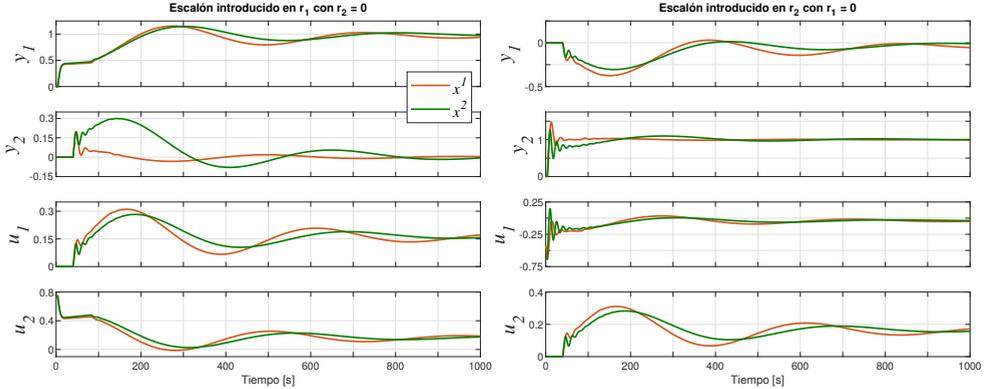
**Figura 5.2:** Conjunto  $P_{Q,n}$  encontrado por nevMOGA. Los controladores óptimos (conjunto  $P_Q$ ) se muestran en naranja y el resto (controladores casi-óptimos no dominados en su vecindad) en verde. Las variables de decisión se visualizan mediante la herramienta LD, utilizando la norma 2 ( $\|\cdot\|_2$ ).

- $Iteraciones = 500$
- $n\_box = [50 \ 50]$

para la definición de los parámetros restantes, se toman los valores sugeridos en [37] para el algoritmo original (ev-MOGA).

La figura 5.2 muestra el conjunto de soluciones  $P_{Q,n}$  encontradas por nevMOGA, donde  $\mathbf{x}^1$ ,  $\mathbf{x}^3$  y  $\mathbf{x}^5$  son soluciones óptimas, y  $\mathbf{x}^2$ ,  $\mathbf{x}^4$  y  $\mathbf{x}^6$  son soluciones casi-óptimas. Para mostrar las variables de decisión, empleamos la herramienta de visualización Level Diagrams (LD), utilizando la norma 2. Con lo que respecta a los objetivos de diseño, al tener solo dos, es preferible analizar su compensación en una sola gráfica. Por este motivo, no hemos empleado la herramienta de visualización LD para la representación de los valores objetivos de los controladores obtenidos. Ahora, fijémonos en las variables de decisión  $x_1$  y  $x_2$  (figura 5.2, derecha). En ellas, se observa cómo nevMOGA ha podido identificar nuevas áreas que contienen soluciones que pueden ser interesantes para el diseñador y que se habrían ignorado en un enfoque de optimización tradicional, donde solo se caracteriza el frente de Pareto  $P_Q$ .

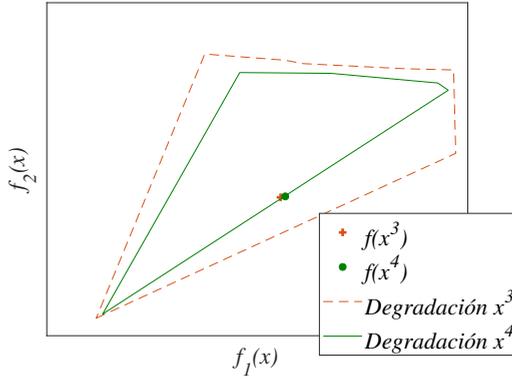
La figura 5.3 muestra las respuestas del sistema para los controladores  $\mathbf{x}^1$  (óptimo) y  $\mathbf{x}^2$  (casi-óptimo), que tienen valores muy similares en  $f_1$  y  $f_2$  (vea la figura 5.2, izquierda). Estas dos soluciones se encuentran en distintos ve-



**Figura 5.3:** Respuesta del sistema para  $\mathbf{x}^1$  (controlador óptimo) y  $\mathbf{x}^2$  (controlador casi-óptimo). Aunque  $\mathbf{f}(\mathbf{x}^2) \simeq \mathbf{f}(\mathbf{x}^1)$ , ambas respuestas transitorias son diferentes, ya que  $\mathbf{x}^1$  y  $\mathbf{x}^2$  son controladores significativamente diferentes.

cindarios y, como era previsible, sus respuestas correspondientes también son diferentes.

Un análisis más profundo de estas dos respuestas transitorias muestra que  $\mathbf{x}^1$  presenta una respuesta mejor que  $\mathbf{x}^2$  con respecto a  $y_2$  cuando el escalón se aplica a  $r_1$ . Sin embargo,  $\mathbf{x}^2$  presenta una mejor respuesta que  $\mathbf{x}^1$  con respecto a  $y_2$  y  $y_1$  (mejor en el sentido de que muestra menor sobreoscilación) cuando el escalón es aplicado a  $r_2$ . Por lo tanto, no está claro en absoluto qué solución se debe elegir. En estos casos, el diseñador deberá incluir nuevos criterios o preferencias, para poder tomar la decisión final. De todos modos, tenga en cuenta que el controlador casi-óptimo  $\mathbf{x}^2$  podría en última instancia ser preferido sobre el controlador óptimo  $\mathbf{x}^1$ . Esto evidencia que no habría sido prudente descartar  $\mathbf{x}^2$  en la etapa de búsqueda y prueba la conveniencia de encontrar las soluciones casi-óptimas además de las óptimas. Esta situación es muy común cuando las funciones objetivo se definen como una agregación de varios objetivos parciales, como es el caso aquí. Además, vale la pena señalar que si este MOP se hubiera abordado con un enfoque tradicional y ocho objetivos (uno para cada término de la ecuación 5.4, sin agregar objetivos) — que es el enfoque ideal si uno quiere encontrar todas las soluciones óptimas, aunque no sea práctico en la etapa de optimización y toma de decisiones —, entonces  $\mathbf{x}^2$  podría haber sido una solución óptima. Esta es una evidencia adicional de la relevancia de  $\mathbf{x}^2$ , que nevMOGA ha podido identificar.



**Figura 5.4:** Límites de degradación de los controladores  $x^3$  (óptimo) y  $x^4$  (casi-óptimo). Variando en  $\pm 20\%$  los valores nominales de la planta, el rendimiento de los controladores  $x^3$  y  $x^4$  se encuentran dentro de los límites obtenidos.

A continuación, evaluamos la robustez de cada controlador encontrado por el algoritmo, es decir, como un controlador mantiene su desempeño cuando se contemplan errores de identificación. Este análisis es típico en el diseño de controladores. Para ello, primero, generamos 50 valores aleatorios para cada parámetro de la planta (ganancias, constantes de tiempo y retardos) dentro de un rango de  $\pm 20\%$ . De este modo, se obtienen 50 variaciones de la planta, que representan 50 desviaciones de la planta nominal. Luego, evaluamos el rendimiento de cada controlador (mediante los objetivos  $f_1$  y  $f_2$ ) para cada una de esas 50 desviaciones de la planta. Incluir un objetivo adicional, que mida de esta forma la robustez y computarlo en la etapa de optimización, junto con los otros dos objetivos, hubiera sido muy costoso computacionalmente (evaluar 50 veces más  $f_1$  y  $f_2$  para todos los controladores generados por el algoritmo en cada iteración, sean o no despreciables). Por este motivo, desplazamos este análisis de robustez a la etapa de toma de decisiones. Como ejemplo de los resultados de este análisis de robustez, la figura 5.4 muestra los límites de la degradación de rendimiento (determinados por los valores de  $f_1$  y  $f_2$  cuando se evalúan para cada uno de las 50 desviaciones de la planta) experimentados por dos controladores ( $x^3$  y  $x^4$ ). Tenga en cuenta que  $x^4$  (controlador casi-óptimo) experimenta menor degradación que  $x^3$ . Esta es una información muy valiosa, y podría hacer que el diseñador opte por  $x^4$  en lugar de  $x^3$ , ya que ambos controladores obtienen rendimiento similar en los objetivos de diseño.

Por último, vemos una última prueba de la conveniencia de considerar las soluciones casi-óptimas. Si comparamos  $x^5$  (controlador óptimo) y  $x^6$  (controlador

casi-óptimo), vemos que tienen un rendimiento similar con respecto a  $f_1$  y  $f_2$ . Sin embargo,  $\mathbf{x}^6$  tiene valores más bajos (en valor absoluto) de las ganancias  $Kc_1$  y  $Kc_2$ . Desde una perspectiva de diseño, este hecho puede ser una ventaja, ya que este controlador presentará menor sensibilidad a cualquier ruido de medición en las salidas de la planta y, por lo tanto, podría preferirse al controlador óptimo  $\mathbf{x}^5$ .

### 5.3 Diseño del control para la columna de destilación Wood & Berry

En esta sección se utiliza la columna de destilación Wood & Berry [108] (ver figura 5.5) como proceso sobre el cual diseñar su sistema de control. La columna de destilación Wood & Berry ha sido estudiada por muchos autores ([59], [71]). Este sistema tiene dos variables manipuladas: flujo de vapor  $V$  y reflujo  $R$  en  $lb/min$ ; y dos salidas: concentración del producto destilado  $x_D$ , y concentración de fondo  $x_B$  por peso de metanol. Además, el flujo de alimentación  $F$  es una perturbación del sistema en  $lb/min$ . El modelo de la planta para el punto de operación ( $x_D = 0.96$ ,  $x_B = 0.02$ ,  $R = 1.95 lb/min$ ,  $S = 1.71 lb/min$  y  $F = 2.45 lb/min$ ) es:

$$Y = \begin{pmatrix} x_D \\ x_B \end{pmatrix} = G(s) \begin{pmatrix} R \\ V \end{pmatrix} + Gd(s) F \quad (5.5)$$

$$G(s) = \begin{pmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{pmatrix}; Gd(s) = \begin{pmatrix} \frac{3.8e^{-8.1s}}{14.9s+1} \\ \frac{4.9e^{-3.4s}}{13.2s+1} \end{pmatrix}$$

donde las constantes de tiempo y los retrasos se miden en minutos.

La estructura de control se define como un emparejamiento diagonal de controladores PI, es decir, la salida  $x_D$  se controla mediante  $R$  y  $x_B$  mediante  $V$ . Esto es:

$$V = Kc_1 \left( e_1(s) + \frac{1}{Ti_1} \frac{1}{s} e_1(s) \right)$$

$$R = Kc_2 \left( e_2(s) + \frac{1}{Ti_2} \frac{1}{s} e_2(s) \right)$$

donde  $Kc_1$  y  $Kc_2$  son las ganancias proporcionales,  $Ti_1$  y  $Ti_2$  son las constantes de tiempo integrales (en minutos) y  $e_1 = sp_{x_D} - x_D$  y  $e_2 = sp_{x_B} - x_B$  son los errores de salida, donde  $sp_{x_D}$  y  $sp_{x_B}$  son las referencias para  $x_D$  y  $x_B$  respectivamente.

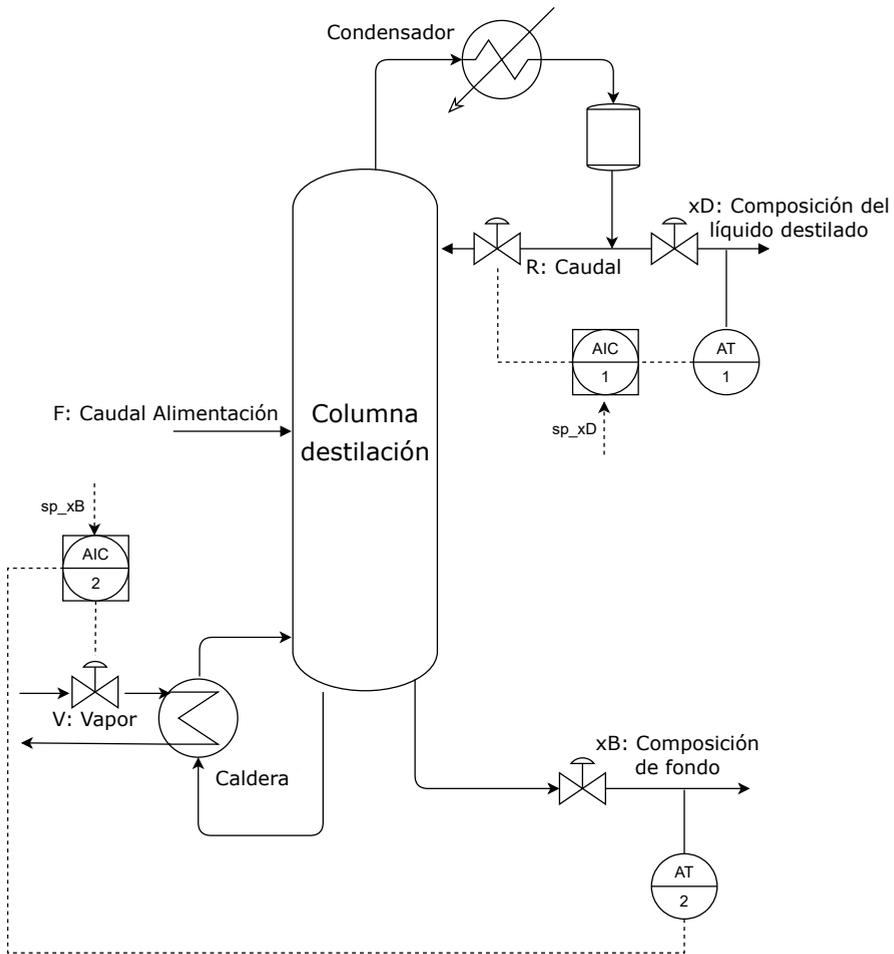
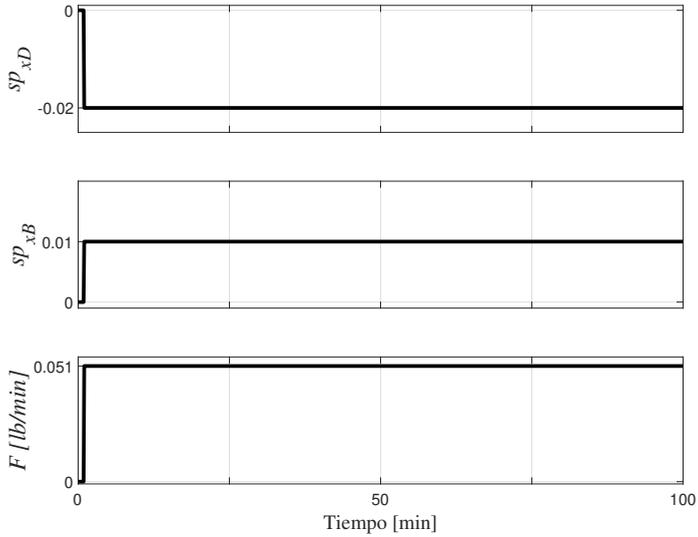


Figura 5.5: Diagrama P&ID de la columna de destilación Wood & Berry [108].



**Figura 5.6:** Escalones con respecto al punto de operación introducidos de forma independiente para el control de la columna de destilación Wood & Berry.

Para analizar el rendimiento de los controladores, se analizan tres situaciones con respecto al punto de operación, de forma independiente: un escalón en  $sp_{xD}$  con  $sp_{xB} = F = 0$ , un escalón en  $sp_{xB}$  con  $sp_{xD} = F = 0$  y un escalón en la perturbación  $F$  con  $sp_{xD} = sp_{xB} = 0$ . Estos tres escalones se muestran en la figura 5.6.

El MOP para la columna de destilación Wood & Berry se define a continuación:

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \quad f_2(\mathbf{x})] \quad (5.6)$$

donde

$$\begin{aligned}
 f_1(\mathbf{x}) &= \sum_{i=11}^{16} f_i \\
 f_2(\mathbf{x}) &= \sum_{i=21}^{26} f_i \\
 \mathbf{x} &= [Kc_1 \quad Ti_1 \quad Kc_2 \quad Ti_2]
 \end{aligned} \tag{5.7}$$

donde

$$f_{11}(\mathbf{x}) = \frac{IAE_1(\mathbf{x})}{IAE_1(\mathbf{x}^R)} \Big|_{\substack{sp_{xD}=1 \\ sp_{xB}=0 \\ F=0}} dt \tag{5.8}$$

$$f_{12}(\mathbf{x}) = \frac{IAE_2(\mathbf{x})}{IAE_2(\mathbf{x}^R)} \Big|_{\substack{sp_{xD}=1 \\ sp_{xB}=0 \\ F=0}} dt \tag{5.9}$$

$$f_{13}(\mathbf{x}) = \frac{IAE_1(\mathbf{x})}{IAE_1(\mathbf{x}^R)} \Big|_{\substack{sp_{xD}=0 \\ sp_{xB}=1 \\ F=0}} dt \tag{5.10}$$

$$f_{14}(\mathbf{x}) = \frac{IAE_2(\mathbf{x})}{IAE_2(\mathbf{x}^R)} \Big|_{\substack{sp_{xD}=0 \\ sp_{xB}=1 \\ F=0}} dt \tag{5.11}$$

$$f_{15}(\mathbf{x}) = \frac{IAE_1(\mathbf{x})}{IAE_1(\mathbf{x}^R)} \Big|_{\substack{sp_{xD}=0 \\ sp_{xB}=0 \\ F=1}} dt \tag{5.12}$$

$$f_{16}(\mathbf{x}) = \frac{IAE_2(\mathbf{x})}{IAE_2(\mathbf{x}^R)} \Big|_{\substack{sp_{xD}=0 \\ sp_{xB}=0 \\ F=1}} dt \tag{5.13}$$

$$f_{21}(\mathbf{x}) = \frac{IAVU_1(\mathbf{x})}{IAVU_1(\mathbf{x}^R)} \Big|_{\substack{sp_{xD}=1 \\ sp_{xB}=0 \\ F=0}} dt \tag{5.14}$$

$$f_{22}(\mathbf{x}) = \frac{IAVU_2(\mathbf{x})}{IAVU_2(\mathbf{x}^R)} \Big|_{\substack{sp_{xD}=1 \\ sp_{xB}=0 \\ F=0}} dt \tag{5.15}$$

$$f_{23}(\mathbf{x}) = \frac{IAVU_1(\mathbf{x})}{IAVU_1(\mathbf{x}^R)} \Big|_{\substack{sp_{xD}=0 \\ sp_{xB}=1 \\ F=0}} dt \tag{5.16}$$

$$f_{24}(\mathbf{x}) = \frac{IAVU_2(\mathbf{x})}{IAVU_2(\mathbf{x}^R)} \Big|_{\substack{sp_{xD}=0 \\ sp_{xB}=1 \\ F=0}} dt \tag{5.17}$$

$$f_{25}(\mathbf{x}) = \frac{IAVU_1(\mathbf{x})}{IAVU_1(\mathbf{x}^R)} \Big|_{F=1}^{sp_{xD}=0, sp_{xB}=0} dt \quad (5.18)$$

$$f_{26}(\mathbf{x}) = \frac{IAVU_2(\mathbf{x})}{IAVU_2(\mathbf{x}^R)} \Big|_{F=1}^{sp_{xD}=0, sp_{xB}=0} dt \quad (5.19)$$

donde

$$\begin{aligned} IAE_i &= \int_0^{100min} |e_i(t)| dt \\ IAVU_i &= \int_0^{100min} \left| \frac{du_i(t)}{dt} \right| dt \end{aligned} \quad (5.20)$$

$$\mathbf{x}^R = [0.6523 \quad 16.7 \quad -0.1237 \quad 14.4]$$

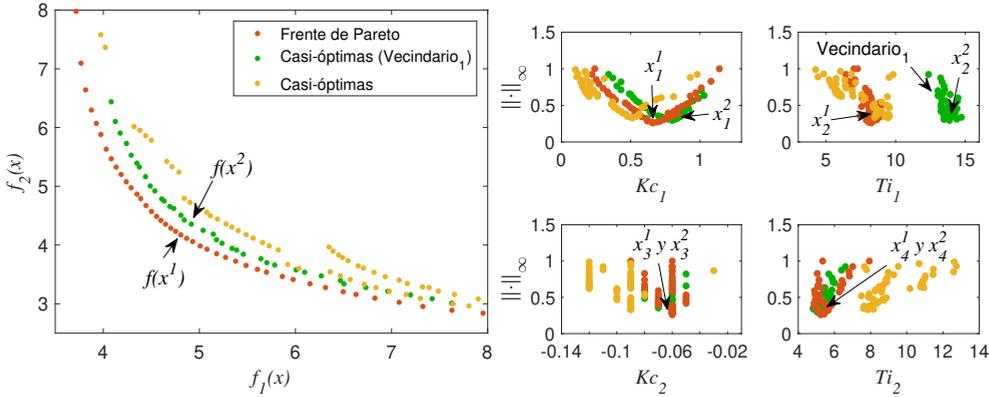
sujeto a

$$\begin{aligned} \mathbf{x} &\leq \mathbf{x} \leq \bar{\mathbf{x}} \\ f_{ij}(\mathbf{x}) &< 2 \quad \forall i = [1, 2], \quad j = [1, \dots, 5] \end{aligned}$$

y donde

$$\begin{aligned} \mathbf{x} &= [0.01 \quad 0.1 \quad -0.5 \quad 0.1] \\ \bar{\mathbf{x}} &= [2 \quad 50 \quad -0.01 \quad 50] \end{aligned}$$

En este MOP, de nuevo, se han agregado diferentes objetivos para simplificar el proceso de optimización y la fase de decisión. Esta agregación se realiza con el mismo peso porque tienen las mismas unidades y la importancia relativa de estos objetivos es la misma. Por lo tanto, tenemos un MOP con dos objetivos de diseño. Para calcular el primer objetivo  $f_1$  la integral del error absoluto (IAE) se agrega sobre ambas salidas, relativizadas sobre el controlador de referencia  $\mathbf{x}^R$  (calculado utilizando la técnica S-IMC [94]), en cada uno de los tres casos mencionados: un escalón en  $sp_{xD}$  ( $f_{11}$  y  $f_{12}$ ), un escalón en  $sp_{xB}$  ( $f_{13}$  y  $f_{14}$ ) y un escalón en la perturbación  $F$  ( $f_{15}$  y  $f_{16}$ ). Del mismo modo, las derivadas de las acciones de control, relativizadas sobre el controlador de referencia ( $\mathbf{x}^R$ ), se miden en cada uno de los tres casos mencionados ( $f_{21}$ ,  $f_{22}$ ,  $f_{23}$ ,  $f_{24}$ ,  $f_{25}$  y  $f_{26}$ ). Estos valores se agregan para formar  $f_2$ . Las variables de decisión del MOP son



**Figura 5.7:** Conjunto de controladores óptimos (naranja) y casi-óptimos no dominados en su vecindario (verde y amarillo) para el MOP de Wood & Berry. A la izquierda, se muestra el valor objetivo de cada una de las opciones encontradas. A la derecha, se muestra las variables de decisión representadas mediante LD, usando la norma infinito.

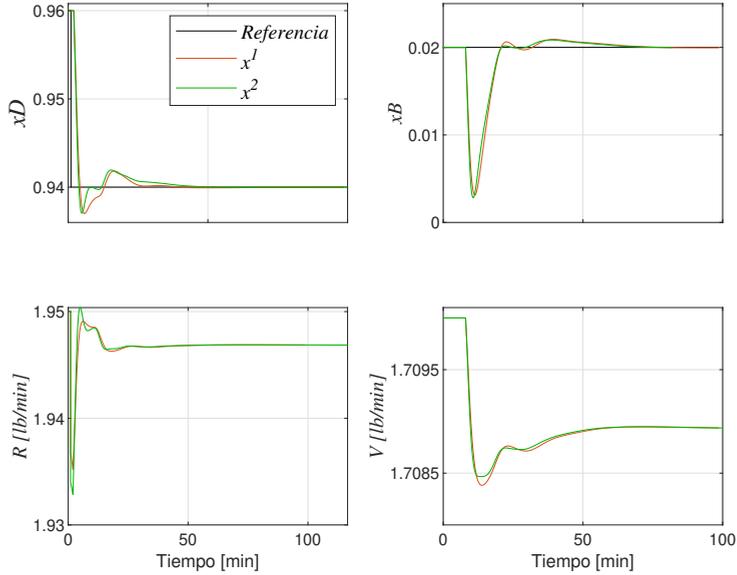
$Kc$  y  $Ti$  para los dos PI propuestos para el control de la planta (ver ecuación 5.3).

Para optimizar el MOP de la columna de destilación Wood & Berry definido en (ecuación 5.6) se utiliza nevMOGA con la siguiente configuración:

- $Nind_G = 8$
- $Nind_P = 250$
- $Iteraciones = 1000$
- $n\_box = [80 \ 80]$
- $\epsilon = [0.5 \ 0.5]$
- $n = [0.25 \ 5 \ 0.025 \ 5]$

para la definición de los parámetros restantes, se toman los valores sugeridos en [37] para el algoritmo original (ev-MOGA).

La figura 5.7 muestra el conjunto  $P_{Q,n}$  de controladores obtenidos por nevMOGA (óptimos y casi-óptimos) para el problema planteado. En primer lugar, se observa un vecindario aislado ( $Vecindario_1$ , soluciones en verde) con valores más altos de  $Ti_1$ . Estas soluciones son las opciones casi-óptimas con rendi-



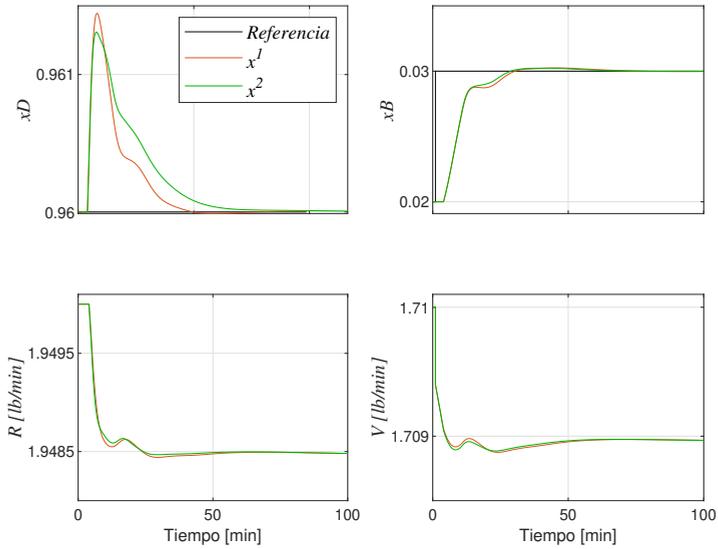
**Figura 5.8:** Respuesta de los controladores  $\mathbf{x}^1$  y  $\mathbf{x}^2$  para la referencia introducida sobre  $x_D$  en la columna de destilación Wood & Berry.

**Tabla 5.1:** Controladores  $\mathbf{x}^1$  y  $\mathbf{x}^2$  para la columna de destilación Wood & Berry.

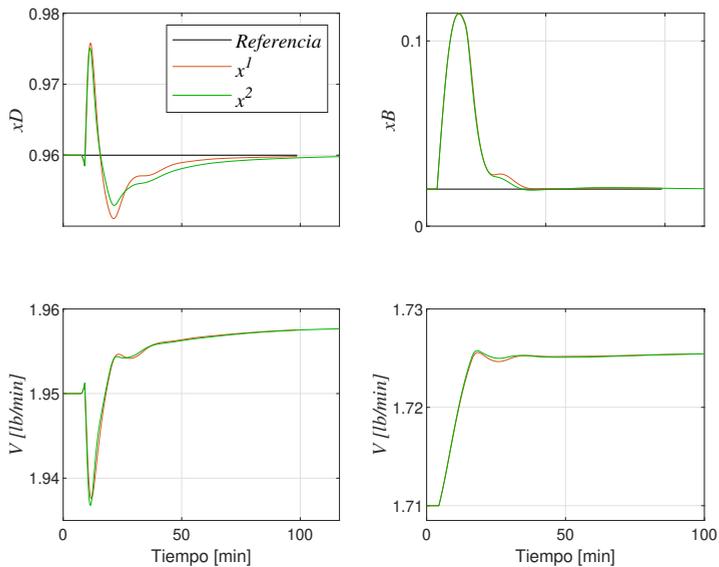
Controlador	$Kc_1$	$Ti_1$	$Kc_2$	$Ti_2$	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$
$\mathbf{x}^1$	0.66	8.25	-0.06	5.24	<b>4.806</b>	<b>4.173</b>
$\mathbf{x}^2$	0.8	13.87	-0.06	5.06	4.916	4.356

mientos más similares al frente de Pareto. Se seleccionan dos soluciones para realizar un análisis más detallado de su comportamiento. En primer lugar, se elige el controlador con la norma infinita inferior del frente de Pareto ( $\mathbf{x}^1$ ).  $\mathbf{x}^1$  es una alternativa compensada y podría ser la elección final del diseñador. En segundo lugar, elegimos una solución ( $\mathbf{x}^2$ ), dominada por  $\mathbf{x}^1$ , del vecindario aislado ( $Vecindario_1$ ) que tiene un rendimiento similar a  $\mathbf{x}^1$  (ver figura 5.1).

Los valores de los objetivos agregados de ambas alternativas se pueden ver en la tabla 5.2. En primer lugar, se puede ver en su respuesta temporal ante un escalón sobre  $x_D$  (figura 5.8) que la solución casi-óptima produce un error ligeramente menor para ambas salidas ( $f_{11}$  y  $f_{12}$ ). Con respecto a las acciones de control, el controlador  $\mathbf{x}^1$  obtiene acciones de control más suaves en  $R$  ( $f_{21}$ )



**Figura 5.9:** Respuesta de los controladores  $x^1$  y  $x^2$  para la referencia introducida sobre  $x_B$  en la columna de destilación Wood & Berry.



**Figura 5.10:** Respuesta de los controladores  $x^1$  y  $x^2$  para la referencia introducida sobre  $F$  en la columna de destilación Wood & Berry.

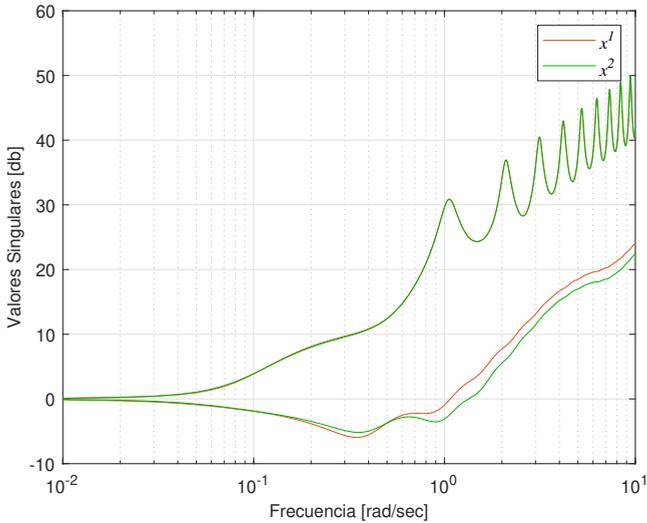
**Tabla 5.2:** Valor objetivo obtenido para los controladores  $\mathbf{x}^1$  y  $\mathbf{x}^2$  para la columna de destilación Wood & Berry.

Controlador	$f_{11}(\mathbf{x})$	$f_{12}(\mathbf{x})$	$f_{13}(\mathbf{x})$	$f_{14}(\mathbf{x})$	$f_{15}(\mathbf{x})$	$f_{16}(\mathbf{x})$
$\mathbf{x}^1$	0.856	1.04	<b>0.499</b>	0.86	<b>0.798</b>	0.751
$\mathbf{x}^2$	<b>0.819</b>	<b>0.973</b>	0.678	<b>0.823</b>	0.894	<b>0.73</b>
Controlador	$f_{21}(\mathbf{x})$	$f_{22}(\mathbf{x})$	$f_{23}(\mathbf{x})$	$f_{24}(\mathbf{x})$	$f_{25}(\mathbf{x})$	$f_{26}(\mathbf{x})$
$\mathbf{x}^1$	<b>0.965</b>	0.417	0.431	0.387	<b>1.147</b>	0.826
$\mathbf{x}^2$	1.186	<b>0.374</b>	<b>0.405</b>	<b>0.381</b>	1.19	<b>0.82</b>

y más bruscas en  $V$  ( $f_{22}$ ) con respecto a  $\mathbf{x}^2$ . En segundo lugar, se observa la diferencia más grande entre los controladores en el seguimiento de referencia ante un escalón sobre  $x_B$  (ver figura 5.9). Mientras que en  $x_B$  ( $f_{14}$ ) el controlador  $\mathbf{x}^2$  tiene un error ligeramente menor, en  $x_D$  ( $f_{13}$ ) es el controlador  $\mathbf{x}^1$  que produce un error significativamente menor. Ambas acciones de control ( $f_{23}$  y  $f_{24}$ ) son más suaves para el controlador  $\mathbf{x}^2$ . Finalmente, se puede ver la respuesta ante un escalón en la perturbación (consulte la figura 5.10). En este caso, el controlador  $\mathbf{x}^1$  obtiene un error menor para  $x_D$  ( $f_{15}$ ) con acciones de control más suaves en  $R$  ( $f_{25}$ ). El controlador  $\mathbf{x}^2$  obtuvo errores más pequeños para  $x_B$  ( $f_{16}$ ) con acciones de control más suaves en  $V$  ( $f_{26}$ ).

Por lo tanto, si nos centramos en el valor de los objetivos optimizados, el controlador  $\mathbf{x}^1$  domina a  $\mathbf{x}^2$ . Pero como se ve en sus respuestas, a pesar de tener valores similares para  $f_1$  y  $f_2$ , ambos controladores tienen características muy diferentes. Después de observar estas respuestas, el diseñador puede priorizar, por ejemplo, obtener un controlador con menor sobreoscilación, lo que favorecería la elección de  $\mathbf{x}^2$  (con una sobreoscilación significativamente menor en  $x_D$  ante el escalón introducido sobre  $sp_{x_B}$ , vea la figura 5.9). Por lo tanto, es muy útil comparar controladores de diferentes vecindarios. Esta comparación detallada nos proporciona nueva información para tomar una decisión final con mayor criterio.

Ahora analizamos un nuevo indicador, relacionado con la robustez, que no está incluido en la fase de optimización. Para el estudio de la robustez de los controladores  $\mathbf{x}^1$  (óptimo) y  $\mathbf{x}^2$  (casi-óptimo), obtenemos los valores mínimos singulares de la incertidumbre multiplicativa [48] [95] [105]. Para hacer esto, observamos la respuesta en frecuencias (ver figura 5.11). El controlador  $\mathbf{x}^2$  tiene un valor mínimo singular de incertidumbre multiplicativa ( $\gamma = 0.55$ ) mayor que  $\mathbf{x}^1$  ( $\gamma = 0.5$ ). Este valor indica que el controlador casi-óptimo  $\mathbf{x}^2$  es más estable (robusto) que el controlador del frente de Pareto  $\mathbf{x}^1$ . Esta característica no se



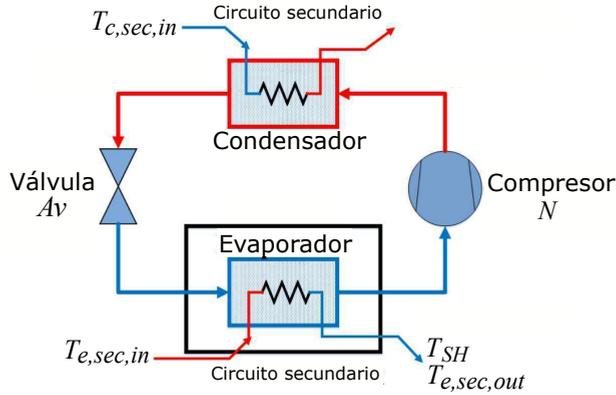
**Figura 5.11:** Valor singular de incertidumbre multiplicativa obtenido para los controladores  $x^1$  y  $x^2$ .

ha tenido en cuenta en los objetivos de diseño, pero podría cambiar la elección hacia la solución casi-óptima  $x^2$ , ya que ambas tienen un rendimiento muy similar en los objetivos de diseño.

Por lo tanto, en este problema de diseño de un controlador para la planta Wood & Berry, se ha puesto de manifiesto la utilidad, para el diseñador, de obtener los controladores casi-óptimos no dominados en su vecindario. Con ellos, se puede hacer un estudio detallado que permita al diseñador elegir con mayor criterio una opción óptima o casi-óptima, ya que ambos pueden tener un desempeño similar con respecto a los objetivos de diseño.

## 5.4 Diseño del control de un sistema de refrigeración (concurso CIC2018)

En esta sección se diseña un control para el sistema de refrigeración propuesto por el Concurso de Ingeniería de Control 2018 (CIC2018 [7]). Los sistemas de refrigeración son, como se conoce generalmente, ciclos cerrados, cuyos componentes están conectados a través de varias tuberías y válvulas, lo que provoca fuertes no linealidades y un alto acoplamiento. Esta es la razón por la que su modelado dinámico no es una cuestión trivial. Por un lado, los elementos más



**Figura 5.12:** Diagrama del sistema de refrigeración. Imagen proporcionado por el *benchmark* de control CIC2018 [7].

importantes son los intercambiadores de calor. Por otro lado, la válvula de expansión, el compresor y el comportamiento térmico de los flujos secundarios se modelan de forma estática. Habitualmente, se opera el ciclo con un cierto grado de sobrecalentamiento del refrigerante en la salida del evaporador, que se mantiene bajo para aproximarse al comportamiento ideal [7] (el refrigerante en la salida del evaporador es vapor saturado).

El CIC2018 propone el control de dos salidas del proceso: la temperatura de salida del fluido secundario del evaporador  $T_{e,sec,out}$  y el grado de sobrecalentamiento  $T_{SH}$  en  $^{\circ}C$ . Además, existen dos variables manipuladas: apertura de la válvula de expansión  $Av$  en (%) y velocidad del compresor  $N$  en  $Hz$ . También hay dos perturbaciones: la temperatura de entrada del fluido secundario del evaporador ( $T_{e,sec,in}$ ) y del condensador ( $T_{c,sec,in}$ ) en  $^{\circ}C$  (vea la figura 5.12).

La estructura de control se define como un control PI que utiliza el siguiente emparejamiento: la salida  $T_{e,sec,out}$  se controla mediante  $Av$  y  $T_{SH}$  por  $N$ . Es decir,

$$\begin{aligned} Av &= Gr_1(s) e_1(s) = Kc_1 \left( e_1(s) + \frac{1}{Ti_1} \frac{1}{s} e_1(s) \right) \\ N &= Gr_2(s) e_2(s) = Kc_2 \left( e_2(s) + \frac{1}{Ti_2} \frac{1}{s} e_2(s) \right) \end{aligned} \quad (5.21)$$

donde  $Kc_1$  y  $Kc_2$  son las ganancias proporcionales,  $Ti_1$  y  $Ti_2$  son las constantes de tiempo integrales en segundos y  $e_1 = sp_{T_{sec}} - T_{e,sec,out}$  y  $e_2 = sp_{T_{SH}} - T_{SH}$  son los errores de salida, donde  $sp_{T_{sec}}$  y  $sp_{T_{SH}}$  son las referencias para  $T_{e,sec,out}$

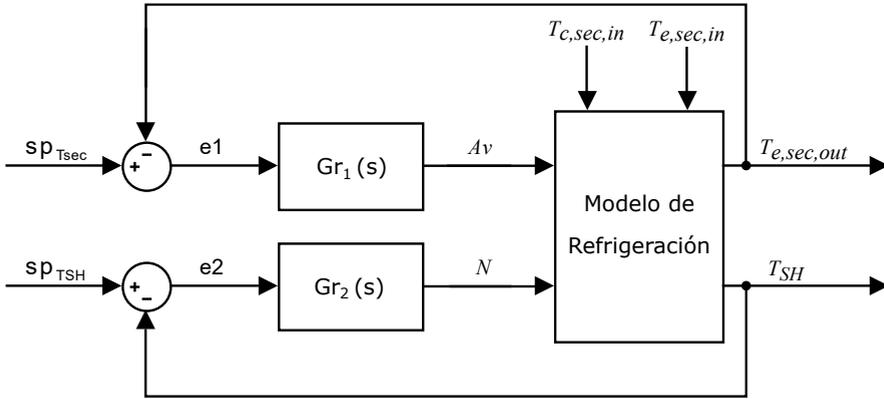


Figura 5.13: Sistema de control para el *benchmark* de control CIC2018 [7].

y  $T_{SH}$  respectivamente. Para la primera fase del concurso, se proporcionan cuatro perfiles: dos para las referencias  $sp_{T_{sec}}$  y  $sp_{T_{SH}}$  (ver figura 5.14) y dos para las perturbaciones  $T_{e,sec,in}$  y  $T_{c,sec,in}$  (vea la figura 5.15). Además, se define el punto de operación (ver tabla 5.3).

#### 5.4.1 Problema de optimización con un único objetivo

En primer lugar, el problema se define con un solo objetivo de diseño. Este objetivo es la agregación de varios objetivos. Esto simplifica el proceso de optimización y la fase de decisión. Para el cálculo de estos objetivos agregados, comenzamos con un controlador de referencia  $x^R$  proporcionado por el concurso que está definido por  $Gr_1$ , controlador de la salida  $T_{e,sec,out}$ , y  $Gr_2$ , controlador de la salida  $T_{SH}$ :

$$Gr_1(s) = \frac{-0.6000s^2 - 0.0300s + 0.5000}{s^2 - 1.9853s + 0.9853} \quad (5.22)$$

$$Gr_2(s) = \frac{0.238s + 0.25}{0.95s}$$

El problema de optimización se define a continuación:

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad (5.23)$$

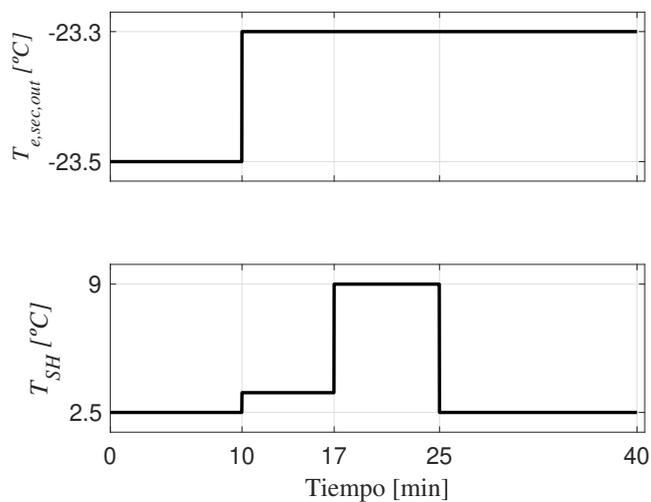


Figura 5.14: Perfil introducido sobre  $sPT_{sec}$  y  $sPT_{SH}$  para el *benchmark* de control CIC2018.

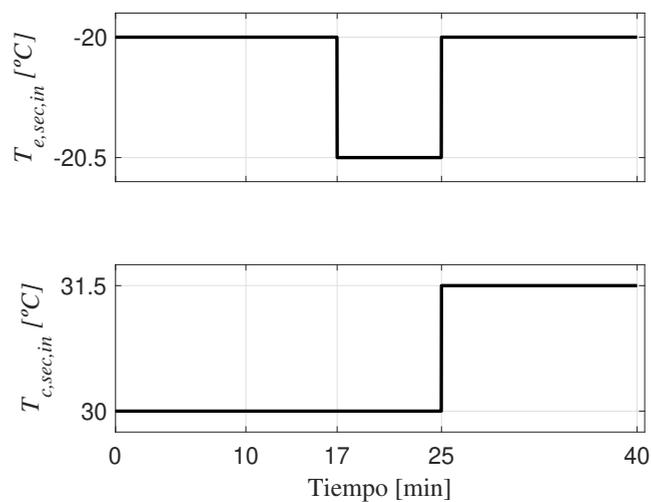


Figura 5.15: Perfil introducido sobre las perturbaciones  $T_{e,sec,in}$  y  $T_{c,sec,in}$  para el *benchmark* de control CIC2018.

**Tabla 5.3:** Punto de operación para el *benchmark* de control CIC2018.

Variable		Valor	Unidades
Variables Manipuladas	$Av$	$\cong 74.45$	%
	$N$	$\cong 42.45$	$Hz$
Perturbaciones	$T_{c,sec,in}$	30	$^{\circ}C$
	$m_{c,sec}$	150	$g s^{-1}$
	$P_{c,sec,in}$	1	$bar$
	$T_{e,sec,in}$	-20	$^{\circ}C$
	$m_{e,sec}$	64.503	$g s^{-1}$
	$P_{e,sec,in}$	1	$bar$
	$T_{surr}$	25	$^{\circ}C$
Variables Controladas	$T_{e,sec,out}$	$\cong -23.5$	$^{\circ}C$
	$T_{SH}$	$\cong 2.5$	$^{\circ}C$

donde

$$f(\mathbf{x}) = \sum_{i=11}^{18} f_i \quad (5.24)$$

$$\mathbf{x} = [Kc_1 \quad Ti_1 \quad Kc_2 \quad Ti_2]$$

donde

$$\begin{aligned} f_{11}(\mathbf{x}) &= w_1 \cdot RIAE_1(\mathbf{x}, x^R) \\ f_{12}(\mathbf{x}) &= w_2 \cdot RIAE_2(\mathbf{x}, x^R) \\ f_{13}(\mathbf{x}) &= w_3 \cdot RITAE_1(\mathbf{x}, x^R, t_{c1}, t_{s1}) \\ f_{14}(\mathbf{x}) &= w_4 \cdot RITAE_2(\mathbf{x}, x^R, t_{c2}, t_{s2}) \\ f_{15}(\mathbf{x}) &= w_5 \cdot RITAE_2(\mathbf{x}, x^R, t_{c3}, t_{s3}) \\ f_{16}(\mathbf{x}) &= w_6 \cdot RITAE_2(\mathbf{x}, x^R, t_{c4}, t_{s4}) \\ f_{17}(\mathbf{x}) &= w_7 \cdot RIAVU_1(\mathbf{x}, x^R) \\ f_{18}(\mathbf{x}) &= w_8 \cdot RIAVU_2(\mathbf{x}, x^R) \end{aligned} \quad (5.25)$$

donde

$$\begin{aligned}
 IAE_i &= \int_0^{40min} |e_i(t)| dt \\
 ITAE_i &= \int_{tc_i}^{tc_i+ts_i} (t - tc_i) |e_i(t)| dt \\
 IAVU_i &= \int_0^{40min} \left| \frac{du_i(t)}{dt} \right| dt
 \end{aligned} \tag{5.26}$$

$$\begin{aligned}
 RIAE_i(\mathbf{x}, x^R) &= \frac{IAE_i(\mathbf{x})}{IAE_i(x^R)} \\
 RITAE_i(\mathbf{x}, x^R, tc_i, ts_i) &= \frac{ITAE_i(\mathbf{x}, tc_i, ts_i)}{ITAE_i(x^R, tc_i, ts_i)} \\
 RIAVU_i(\mathbf{x}, x^R) &= \frac{IAVU_i(\mathbf{x})}{IAVU_i(x^R)}
 \end{aligned} \tag{5.27}$$

sujeto a:

$$\begin{aligned}
 \underline{\mathbf{x}} &\leq \mathbf{x} \leq \bar{\mathbf{x}} \\
 f(\mathbf{x}) &< 0.2
 \end{aligned}$$

y donde

$$\begin{aligned}
 \underline{\mathbf{x}} &= [-35 \ 0.75 \ 0.25 \ 1.75] \\
 \bar{\mathbf{x}} &= [-5 \ 1.5 \ 1.25 \ 3.75]
 \end{aligned}$$

$$\begin{aligned}
 w_1 &= 0.229 & w_2 &= 0.120 & w_3 &= 0.224 & w_4 &= 0.114 \\
 w_5 &= 0.114 & w_6 &= 0.113 & w_7 &= 0.0264 & w_8 &= 0.0593 \\
 tc_1 &= tc_2 = 10 \text{ min} & ts_1 &= ts_2 = tc_3 &= 17 \text{ min} \\
 ts_3 &= tc_4 = 25 \text{ min} & ts_4 &= 40 \text{ min}
 \end{aligned}$$

Se evalúan ocho objetivos de rendimiento agregados para el cálculo del objetivo de diseño. Los dos primeros objetivos agregados son la integral de los errores absolutos (IAE) en ambas variables controladas ( $T_{e,sec,out}$  y  $T_{SH}$ ). El tercero es el tiempo integral del error absoluto (ITAE) para la primera variable controlada ( $T_{e,sec,out}$ ) evaluada después del único escalón introducido sobre ella (en

**Tabla 5.4:** Controladores  $\mathbf{x}^1$  y  $\mathbf{x}^2$  para el enfoque con un único objetivo del control del *benchmark* CIC2018.

Controlador	$Kc_1$	$Ti_1$	$Kc_2$	$Ti_2$	$f(\mathbf{x})$
$\mathbf{x}^1$	-27.7	1.07	0.87	3.21	<b>0.162</b>
$\mathbf{x}^2$	-16.14	1.3	0.53	3.52	0.186

10 *min*, ver figura 5.14). Los objetivos agregados cuarto, quinto y sexto son el ITAE para la segunda variable controlada ( $T_{SH}$ ) evaluada después de los tres escalones introducidos en 10, 17 y 25 *min* (ver figura 5.14). Los objetivos agregados séptimo y octavo son la integral de la variación absoluta de la señal de control (IAVU) para las dos variables manipuladas ( $Av$  y  $N$ ). El objetivo general a optimizar se obtiene utilizando el valor promedio de los ocho objetivos agregados individuales con un factor de ponderación  $w_i$  (propuesto por el *benchmark*) para cada objetivo agregado relativizados sobre el controlador de referencia.

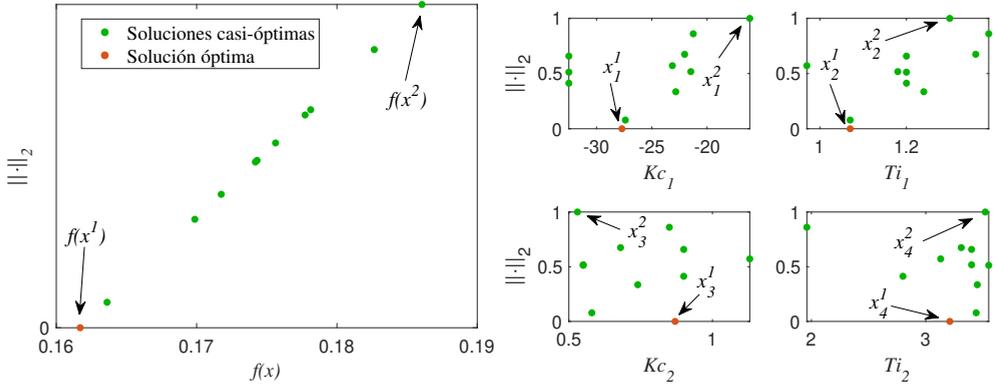
Para optimizar el problema definido en (ecuación 5.23) nevMOGA se utiliza con la siguiente configuración:

- $Nind_P = 250$
- $Nind_G = 4$
- $Iteraciones = 800$
- $n\_box = 100$
- $\epsilon = [0.025]$
- $\mathbf{n} = [5 \quad 0.1 \quad 0.25 \quad 1]$

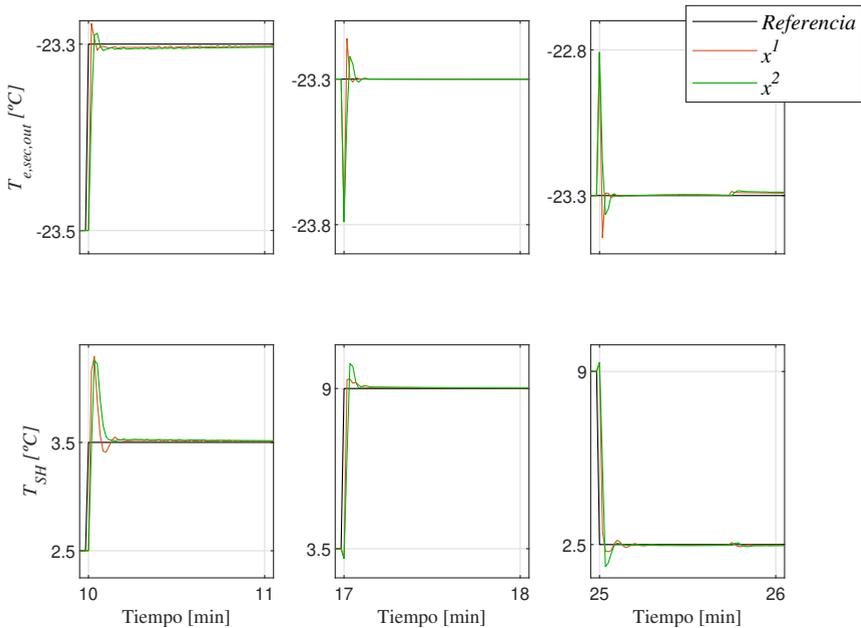
para la definición de los parámetros restantes se toman los valores por defecto sugeridos en [37] para el algoritmo original ev-MOGA.

En primer lugar, se obtienen los controlador óptimo y los controladores casi-óptimos no dominados en su vecindad ( $P_{Q,n}$ ), como se muestra en la figura 5.16.

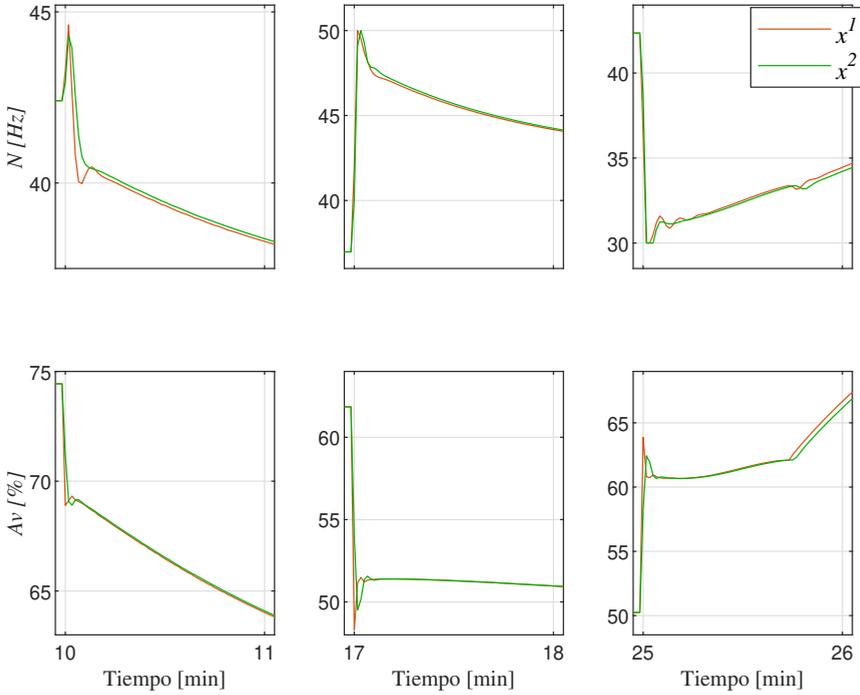
Ahora, seleccionamos el controlador óptimo  $\mathbf{x}^1$  junto con el controlador casi-óptimo  $\mathbf{x}^2$ , que se encuentra en un vecindario diferente. En las figuras 5.17 y 5.18 se observa la respuesta de ambos controles. Además, sus objetivos agrega-



**Figura 5.16:** Controlador óptimo (naranja) y controladores casi-óptimos no dominados en su vecindad (verde), para el *benchmark* CIC2018, visualizados mediante LD, usando la norma 2. A la derecha, se observa el objetivo de diseño. A la izquierda, se muestran las variables de decisión.



**Figura 5.17:** Comparación de las respuestas de los controladores  $x^1$  y  $x^2$  y para el *benchmark* de control CIC2018. Detalles de los momentos en que se produjeron cambios de referencia.



**Figura 5.18:** Comparación de las acciones de control de los controladores  $x^1$  y  $x^2$  para el *benchmark* de control CIC2018. Detalle de los momentos en los que ocurrieron cambios en la referencia.

**Tabla 5.5:** Valor de los objetivos agregados para los controladores  $x^1$  y  $x^2$

Controlador	$f_{11}(\mathbf{x})$	$f_{12}(\mathbf{x})$	$f_{13}(\mathbf{x})$	$f_{14}(\mathbf{x})$
$x^1$	<b>0.026</b>	<b>0.013</b>	<b>0.019</b>	<b>0.005</b>
$x^2$	0.033	0.017	0.027	0.008
Controlador	$f_{15}(\mathbf{x})$	$f_{16}(\mathbf{x})$	$f_{17}(\mathbf{x})$	$f_{18}(\mathbf{x})$
$x^1$	<b>0.004</b>	<b>0.007</b>	0.029	0.059
$x^2$	0.007	0.011	<b>0.027</b>	<b>0.056</b>

dos y el valor objetivo global se muestran en las tablas 5.4 y 5.5. El controlador  $\mathbf{x}^1$  tiene un error menor en las salidas como se muestra en la figura 5.17 y el valor de los objetivos  $f_{11}$  a  $f_{16}$ . El controlador  $\mathbf{x}^2$  obtiene acciones de control más suaves como se observa en la figura 5.18 y los objetivos  $f_{17}$  a  $f_{18}$ . Esto revela el conflicto entre los objetivos agregados en relación con los errores en las salidas y los objetivos relativos al esfuerzo de control. Este conflicto ha sido posible detectarlo gracias a encontrar el conjunto de opciones casi-óptimas no dominadas en su vecindad. Para analizar este conflicto con mayor detalle, se plantea un problema de optimización multiobjetivo con dos objetivos, donde los objetivos agregados que entran en conflicto se agrupan de forma independiente. De este modo, es posible analizar la compensación de los controladores en los objetivos planteados. Los resultados de este enfoque multiobjetivo se discuten en la siguiente sección.

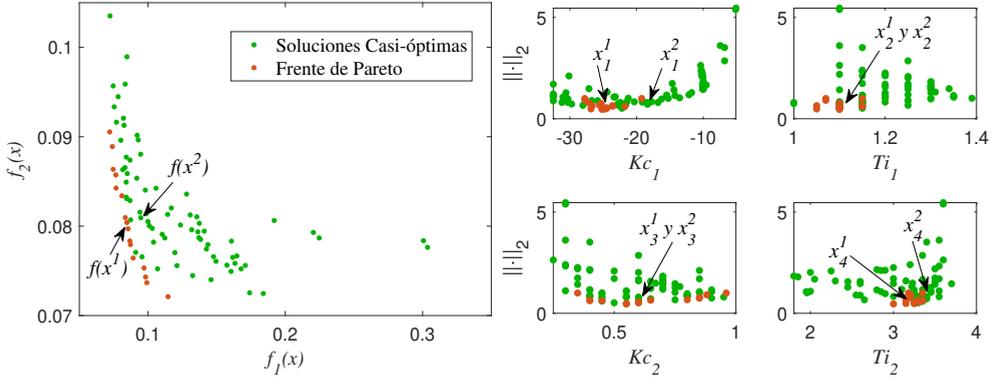
#### 5.4.2 Problema de Optimización Multiobjetivo

En esta sección se define un problema de optimización multiobjetivo para el *benchmark* de control CIC2018. Para ello, se desagrega el objetivo definido en la sección anterior, planteando dos objetivos de diseño. El primer objetivo está formado por los primeros seis objetivos agregados definidos en la ecuación 5.25 con sus factores de ponderación  $w_i$ . El segundo objetivo está formado por los dos últimos objetivos agregados con sus correspondientes factores de ponderación. De esta forma, los objetivos agregados relacionados con el desempeño y el esfuerzo de control, que entran en conflicto, se estudian de forma independiente. El MOP para el *benchmark* de control CIC2018 es:

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \quad f_2(\mathbf{x})] \quad (5.28)$$

donde

$$\begin{aligned} f_1(\mathbf{x}) &= \sum_{i=11}^{16} f_i \\ f_2(\mathbf{x}) &= \sum_{i=17}^{18} f_i \\ \mathbf{x} &= [Kc_1 \quad Ti_1 \quad Kc_2 \quad Ti_2] \end{aligned} \quad (5.29)$$



**Figura 5.19:** Conjunto de controladores óptimos (naranja) y casi-óptimos no dominados en su vecindad (verde) para el *benchmark* de control CIC2018. A la izquierda, se muestra la compensación de los objetivos de diseño. A la derecha, se muestra las variables de decisión representadas con LD, usando la norma 2.

sujeto a:

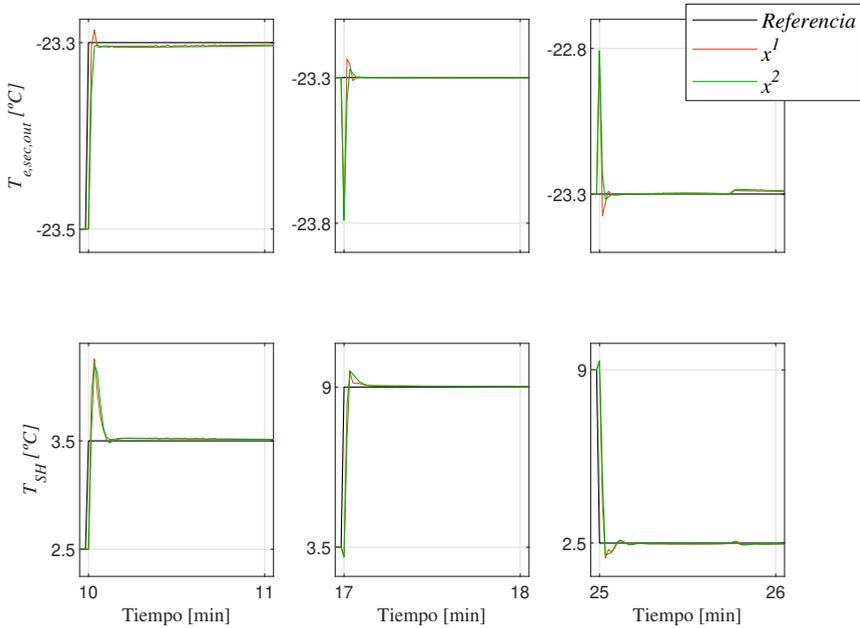
$$\underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}}$$

$$f_i(\mathbf{x}) < 1.4 \quad \forall i \in [1 \ 2]$$

donde  $f_{11}(\mathbf{x}) \dots f_{18}(\mathbf{x})$ ,  $\underline{\mathbf{x}}$ ,  $\bar{\mathbf{x}}$ ,  $w_i$ ,  $t_{ci}$  y  $t_{si}$  están definidos en las ecuaciones 5.25 y 5.26 y  $x^R$  es definido en la ecuación 5.22.

Para la optimización del MOP definido en la ecuación 5.28 nevMOGA se utiliza con la siguiente configuración:

- $Nind_P = 100$
- $Nind_G = 4$
- $Iteraciones = 800$
- $\mathbf{n\_box} = [100 \ 100]$
- $\boldsymbol{\epsilon} = [0.0125 \ 0.0125]$
- $\mathbf{n} = [2.5 \ 0.2 \ 0.25 \ 0.5]$



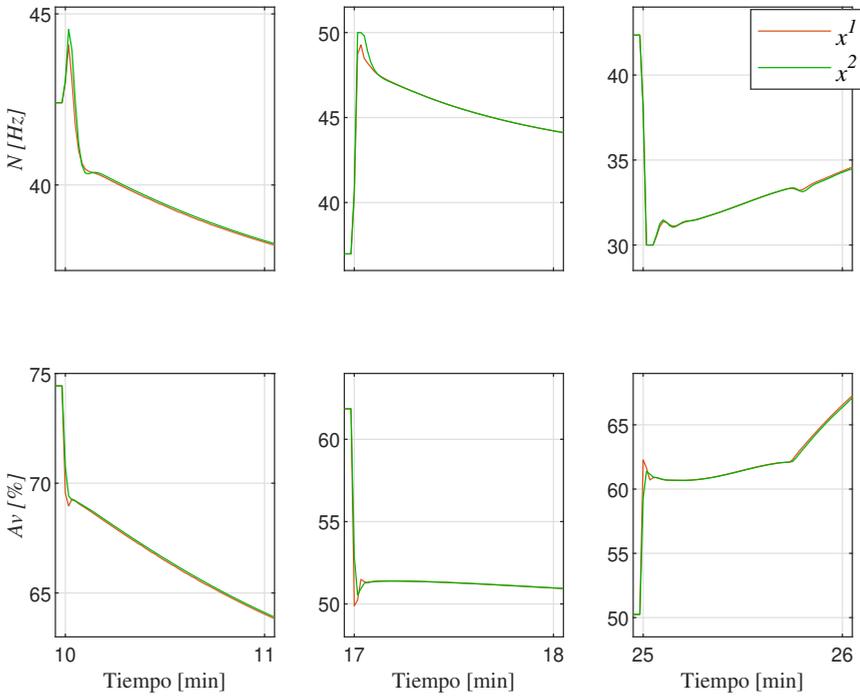
**Figura 5.20:** Comparación de la respuesta de los controladores  $x^1$  y  $x^2$  para el *benchmark* de control CIC2018. Detalle de los instantes de tiempo donde se producen cambios en las referencias.

para la definición de los parámetros restantes se toman los valores predefinidos en [37] para el algoritmo original (ev-MOGA).

La figura 5.19 muestra el conjunto  $P_{Q,n}$  obtenido, utilizando nevMOGA, para el problema propuesto. Dos soluciones son nuevamente elegidas para su análisis. En primer lugar, se elige una solución óptima de compromiso ( $x^1$ ). En segundo lugar, se selecciona una solución casi-óptima  $x^2$  dominada por  $x^1$  en un vecindario diferente (consulte las tablas 5.6 y 5.7, y la figura 5.19) ya que  $K_{C_1}$  es significativamente diferente.

La respuesta de ambos controladores se observa en las figuras 5.20 y 5.21. El controlador  $x^1$  muestra un IAE más bajo en ambas salidas ( $f_{11}$  y  $f_{12}$ ). También muestra mejor ITAE sobre  $T_{e,seg,out}$  ( $f_{13}$ ). La solución casi-óptima ( $x^2$ ) muestra mejor ITAE sobre la segunda salida  $T_{SH}$  ( $f_{14}$  a  $f_{16}$ ).

La acción de control  $Av$  de  $x^1$  es más suave ( $f_{18}$ ), mientras que la acción de control  $N$  es más brusca ( $f_{17}$ ). Por lo tanto,  $x^2$  gana en cuatro ( $f_{14}$  a  $f_{17}$ ) y



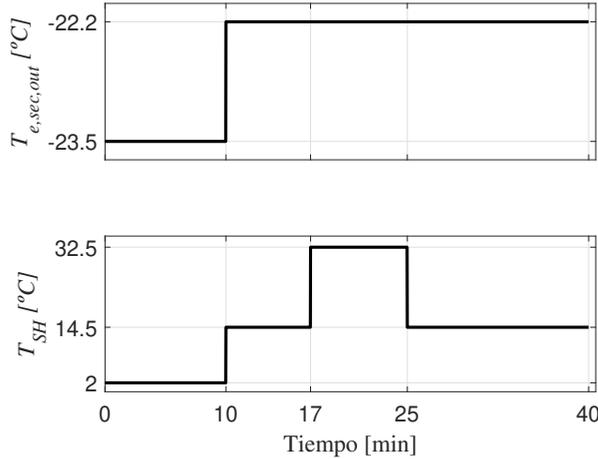
**Figura 5.21:** Comparación de las acciones de control de los controladores  $x^1$  y  $x^2$  para *benchmark* de control CIC2018. Detalle de los instantes en el tiempo donde se producen cambios en las referencias.

**Tabla 5.6:** Valores de los objetivos agregados de los controladores  $x^1$  y  $x^2$ .

Controlador	$f_{11}(\mathbf{x})$	$f_{12}(\mathbf{x})$	$f_{13}(\mathbf{x})$	$f_{14}(\mathbf{x})$
$x^1$	<b>0.027</b>	<b>0.015</b>	<b>0.021</b>	0.007
$x^2$	0.03	0.015	0.028	<b>0.007</b>
Controlador	$f_{15}(\mathbf{x})$	$f_{16}(\mathbf{x})$	$f_{17}(\mathbf{x})$	$f_{18}(\mathbf{x})$
$x^1$	0.006	0.01	0.027	<b>0.054</b>
$x^2$	<b>0.005</b>	<b>0.009</b>	<b>0.024</b>	0.056

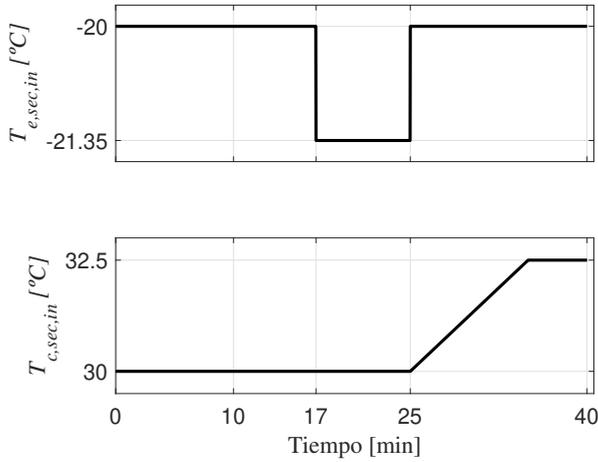
**Tabla 5.7:** Controladores  $\mathbf{x}^1$  y  $\mathbf{x}^2$  obtenidos en el MOP para el *benchmark* de control CIC2018.

Controlador	$Kc_1$	$Ti_1$	$Kc_2$	$Ti_2$	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$
$\mathbf{x}^1$	-24.45	1.1	0.6	3.25	<b>0.085</b>	<b>0.08</b>
$\mathbf{x}^2$	-18.35	1.1	0.65	3.4	0.095	0.081

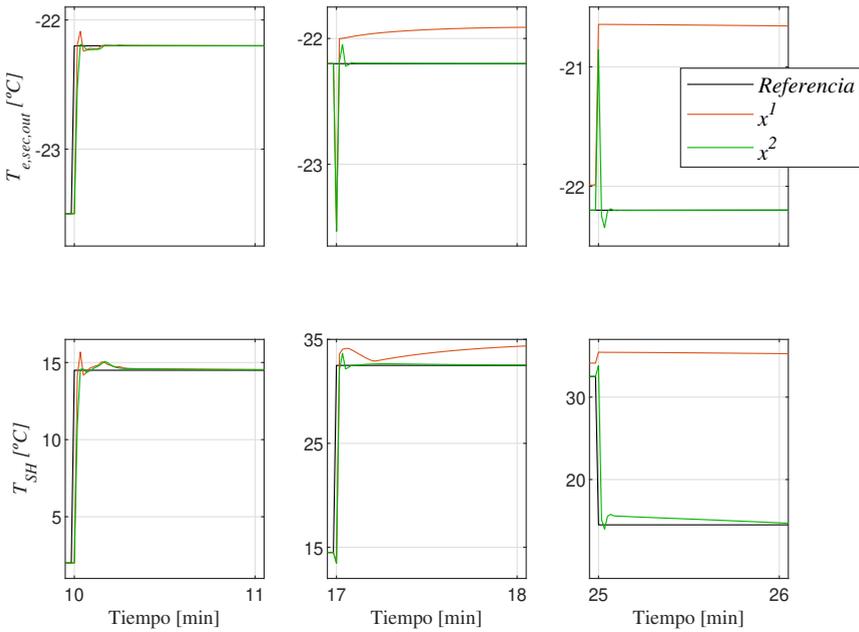
**Figura 5.22:** Escalones en  $sp_{T_{sec}}$  y  $sp_{T_{SH}}$ , introducidos en  $T_{e,sec,out}$  y  $T_{SH}$  respectivamente, para el ensayo de validación del *benchmark* de control CIC2018.

pierde en cuatro ( $f_{11}$  a  $f_{13}$  y  $f_{18}$ ) objetivos agregados. Se puede ver que hay dos controladores con rendimientos similares y, a pesar de que  $\mathbf{x}^2$  está dominado por  $\mathbf{x}^1$ , no se observa una pérdida significativa en ninguno de los objetivos agregados propuestos. Por ello, siendo dos controladores con características diferentes, parece razonable que sea el diseñador, y no el algoritmo, el que seleccione uno de ellos en base a sus especificaciones.

El concurso CIC2018 también propone una segunda fase en la que se proporcionan cuatro nuevos perfiles de validación para las referencias  $sp_{T_{sec}}$  y  $sp_{T_{SH}}$  y las perturbaciones  $T_{e,sec,in}$  y  $T_{c,sec,in}$  (consulte las figuras 5.22 y 5.23). De este modo, se puede comparar el rendimiento de los controladores, seleccionados con el perfil inicial, en un nuevo punto de operación. Esto implica un cambio de escenario para el problema planteado en la primera fase. De esta forma, evaluamos en simulación la robustez de los controladores comparados, analizando un nuevo escenario.



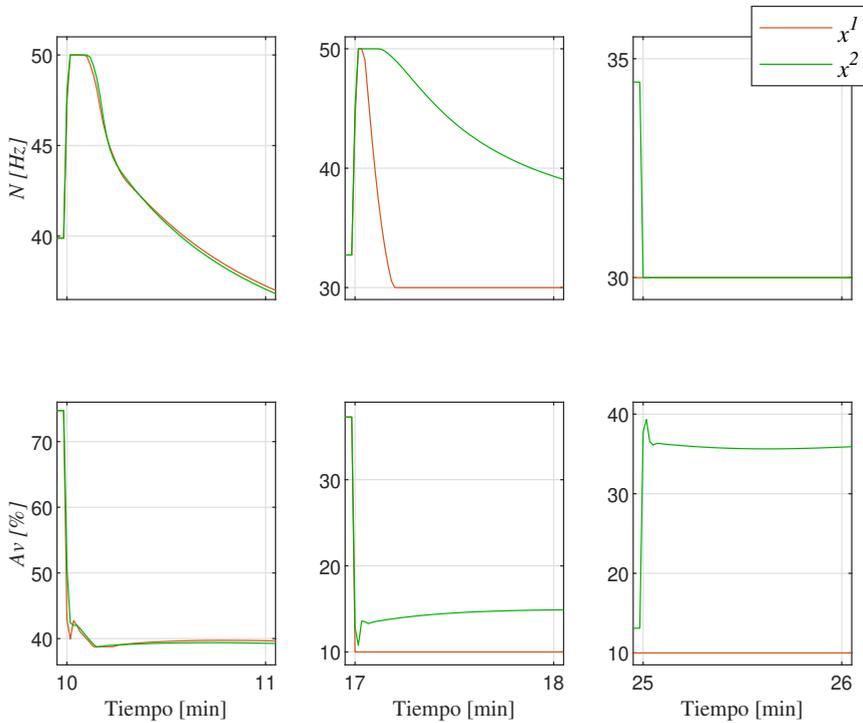
**Figura 5.23:** Escalones sobre las perturbaciones  $T_{e,sec,in}$  y  $T_{c,sec,in}$  para el ensayo de validación del *benchmark* de control CIC2018.



**Figura 5.24:** Comparación de la respuesta para el ensayo de validación de los controladores  $x^1$  y  $x^2$  para el *benchmark* de control CIC2018. Detalle de los instantes donde se producen cambios en las referencias.

**Tabla 5.8:** Los valores de los objetivos de diseño y los objetivos agregados para los controladores  $\mathbf{x}^1$  y  $\mathbf{x}^2$  en la prueba de validación.

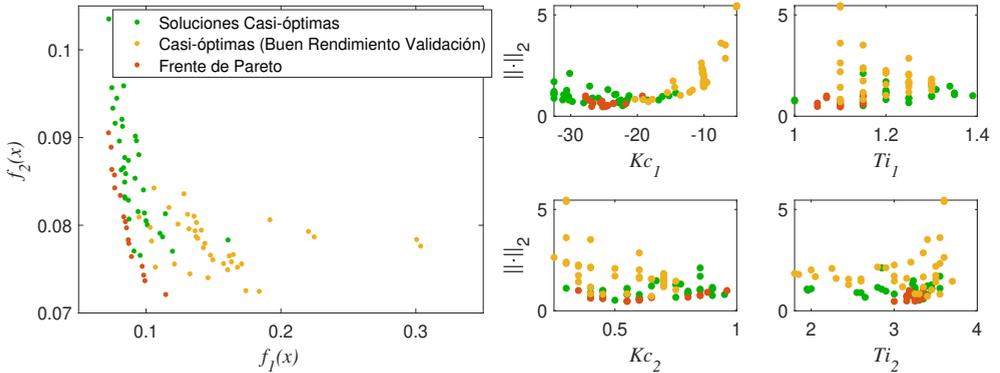
Controlador	$f_{11}(\mathbf{x})$	$f_{12}(\mathbf{x})$	$f_{13}(\mathbf{x})$	$f_{14}(\mathbf{x})$	$f_1(\mathbf{x})$
$\mathbf{x}^1$	3.7361	2.2980	0.0065	0.0053	38.6
$\mathbf{x}^2$	<b>0.0156</b>	<b>0.0150</b>	<b>0.0042</b>	<b>0.0036</b>	<b>0.056</b>
Controlador	$f_{15}(\mathbf{x})$	$f_{16}(\mathbf{x})$	$f_{17}(\mathbf{x})$	$f_{18}(\mathbf{x})$	$f_2(\mathbf{x})$
$\mathbf{x}^1$	1.7415	30.8361	<b>0.0120</b>	<b>0.0282</b>	<b>0.04</b>
$\mathbf{x}^2$	<b>0.0037</b>	<b>0.0134</b>	0.0173	0.0336	0.051



**Figura 5.25:** Comparación de las acciones de control en el ensayo de validación de los controladores  $\mathbf{x}^1$  y  $\mathbf{x}^2$  para el *benchmark* de control CIC2018. Detalle de los instantes donde se producen cambios en las referencias.

En las figuras 5.24 y 5.25, y en la tabla 5.8 se muestra la respuesta y los valores de los objetivos agregados de los controladores  $\mathbf{x}^1$  y  $\mathbf{x}^2$  obtenidos sobre el ensayo de validación. La solución  $\mathbf{x}^2$  muestra un mejor IAE ( $f_{11}$  y  $f_{12}$ ) en este nuevo escenario. Además,  $\mathbf{x}^2$  muestra una mejora en el ITAE para ambas salidas ( $f_{13}$  a  $f_{16}$ ). La diferencia más significativa se observa en el ITAE del último escalón introducido sobre  $T_{SH}$  ( $f_{16}$ ). La diferencia es tal que el controlador  $\mathbf{x}^1$  no alcanza la referencia debido a la saturación de la acción de control en ese intervalo (consulte las figuras 5.24 y 5.25).  $\mathbf{x}^2$  alcanza la referencia y obtiene un rendimiento significativamente mejor en este intervalo. Ambas acciones de control son ligeramente más suaves para  $\mathbf{x}^1$  ( $f_{17}$  y  $f_{18}$ ). Sin embargo, existe una diferencia significativa en el rendimiento (especialmente para  $f_{16}$ , intervalo entre el minuto 25 y 40 en la figura 5.24) de ambos controladores en favor de  $\mathbf{x}^2$ . Por lo tanto, ante dos controladores con un buen rendimiento sobre el ensayo inicial, uno de ellos ( $\mathbf{x}^2$ ) obtiene un rendimiento significativamente mejor sobre el ensayo de validación. Este hecho, decanta nuestra elección hacia un controlador casi-óptimo ( $\mathbf{x}^2$ ) a diferencia de lo que se podría esperar a priori.

Para realizar un estudio más detallado de las soluciones producidas en el MOP propuesto, obtenemos su rendimiento para el ensayo de validación definido (figura 5.26). En este estudio, podemos ver que hay un conjunto de soluciones casi-óptimas (soluciones que se muestran en amarillo) que logran alcanzar el perfil de validación introducido de modo similar a  $\mathbf{x}^2$ . El resto de controladores no consigue un buen desempeño sobre el ensayo de validación. De hecho, ninguna solución óptima alcanza la referencia, tal y como ocurre con  $\mathbf{x}^1$ . Por lo tanto, se muestra la importancia de encontrar soluciones casi-óptimas. Gracias a estas soluciones, hemos podido analizar un nuevo escenario, el ensayo de validación (para la que inicialmente no había información). Este análisis nos ha permitido encontrar controladores casi-óptimos (similares a los óptimos) para los perfiles introducidos inicialmente que obtienen un desempeño significativamente mejor a los controladores óptimos en el ensayo de validación. Con un enfoque multiobjetivo tradicional, este análisis no podría haberse realizado sin redefinir el MOP en el nuevo escenario (y luego obtener los controladores óptimos).



**Figura 5.26:** Conjunto de soluciones óptimas (naranja) y casi-óptimas no dominadas en su vecindad (verde y amarillo) para el perfil inicial introducido en el *benchmark* de control CIC2018. Las soluciones en amarillo (casi-óptimas) obtienen un buen rendimiento en el ensayo de validación.

## 5.5 Identificación del sistema de refrigeración de un sistema $\mu$ -CHP basado en una pila de PEMFC

Actualmente existen problemas medioambientales tales como: calentamiento global, contaminación atmosférica, agotamiento de las reservas de combustibles fósiles, etc. Las nuevas tecnologías en el sector energético pueden ayudar a paliarlos. Este es el caso de la tecnología de los sistemas  $\mu$ -CHP (*Combined Heat and Power* [25] [63]). Un sistema  $\mu$ -CHP genera electricidad y calor para el suministro energético de la vivienda (con potencias menores de 5 kW [2]). Por una lado, la energía eléctrica producida alimenta las cargas eléctricas de la vivienda. Por otro lado, la energía térmica generada se utiliza para la calefacción y el agua caliente. La principal ventaja de estos sistemas es el aprovechamiento de la energía térmica generada durante el proceso de producción de energía eléctrica.

Los sistemas  $\mu$ -CHP utilizan diversas tecnologías ([55], [6]). De entre ellas, algunos autores coinciden en señalar que las más prometedoras, debido a su eficiencia y pocas emisiones, son las tecnologías basadas en la pila de combustible ([2], [29], [36]). Los sistemas  $\mu$ -CHP de este tipo más habituales se basan en pilas de combustible de tipo PEMFC (Proton Exchange Membrane Fuel Cell).

Actualmente existen algunos sistemas  $\mu$ -CHP comerciales basados en PEMFC. Sin embargo, es necesario avanzar en varios aspectos de estos sistema, con el fin de mejorar sus prestaciones y reducir sus costes ([36], [114]). Una de las áreas donde existe un mayor margen de mejora es en el control de temperatura de la PEMFC. El correcto diseño del sistema de refrigeración de la pila es vital en la durabilidad, coste, fiabilidad y eficiencia energética de la pila ([114], [87]) (el componente más costoso del sistema). Sin embargo, para diseñar un buen control de temperatura de la pila es necesario disponer de un adecuado modelo del sistema de refrigeración. Por ello, en esta sección se pretende identificar los parámetros de un modelo de la unidad de refrigeración de un sistema  $\mu$ -CHP.

### 5.5.1 Planteamiento del problema de optimización multiobjetivo

Utilizando el enfoque introducido en la tesis, nevMOGA proporcionará al diseñador, además de los modelos óptimos, los modelos casi-óptimos no dominados en su vecindad. Estos modelos proveerán de un mayor conocimiento al diseñador para tomar la decisión final con mayor criterio. El modelo a identificar (descrito en el Apéndice A) está basado en primeros principios, es dinámico, no lineal. El modelo será ajustado y validado con datos experimentales obtenidos de un proceso real que simula un sistema  $\mu$ -CHP basado en un PEMFC de 2kW. Para llevar a cabo la identificación, desde el punto de vista de la optimización de procesos, partimos de un modelo con un conjunto de parámetros a identificar. Los detalles de este modelo se pueden ver en el Apéndice A. Las entradas del modelo son (representadas en naranja en la figura 5.27):

- $F_a$ : Caudal de aire de la pila en  $m^3/s$
- $T_{amb}$ : Temperatura ambiente en  $^{\circ}C$
- $T_{a_{in}}$ : Temperatura del aire de entrada a la pila en  $^{\circ}C$
- $v$ : Tensión proporcionada por la pila en voltios.
- $i$ : Corriente proporcionada por la pila en amperios
- $F_{w_1}$ : Caudal del circuito primario en  $m^3/s$
- $F_{w_2}$ : Caudal del circuito secundario en  $m^3/s$
- $R$ : Encendido/apagado del radiador en *On/Off*

Las salidas del modelo son (representadas en azul en la figura 5.27):

- $T_{w_{out}}$ : Temperatura de salida del agua de la pila en  $^{\circ}C$



**Figura 5.27:** Entradas y salidas del sistema de refrigeración de un sistema  $\mu$ -CHP.

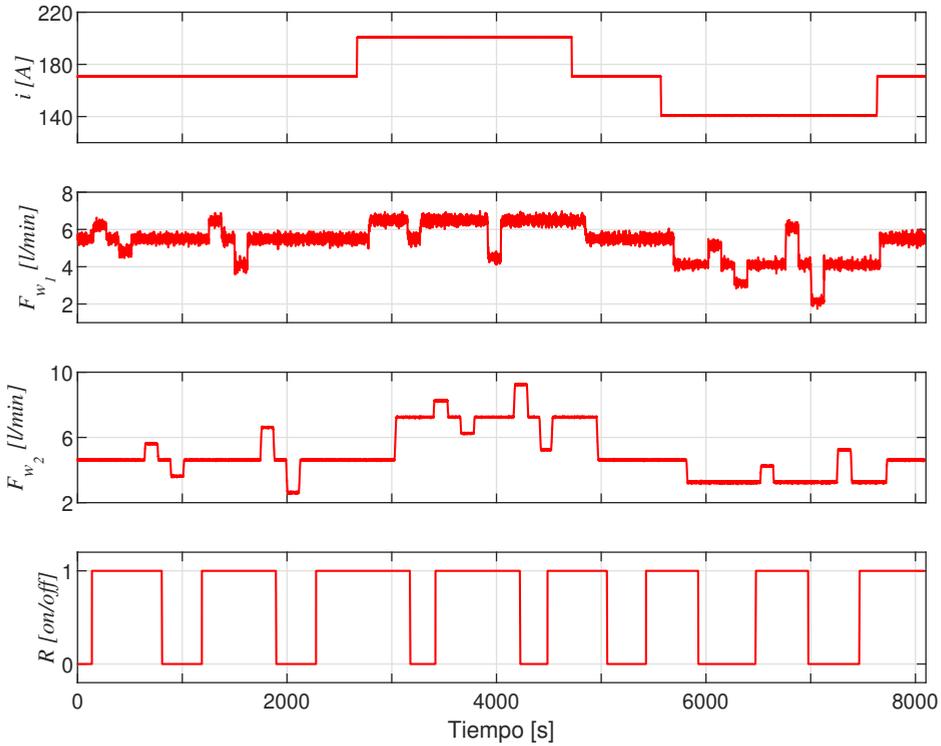
- $T_{w_{in}}$ : Temperatura de entrada del agua de la pila en  $^{\circ}C$
- $T_{t2}$ : Temperatura del interior del depósito 2 en  $^{\circ}C$
- $T_{a_{out}}$ : Temperatura del aire de salida de la pila en  $^{\circ}C$
- $T_{s_{in}}$ : Temperatura del agua a la entrada del intercambiador en la carcasa en  $^{\circ}C$
- $T_{s_{out}}$ : Temperatura del agua a la salida del intercambiador en la carcasa en  $^{\circ}C$

El modelo tiene 30 parámetros a estimar (ver tabla A.2, en el Apéndice A). Para ajustar los parámetros del modelo, se usarán los datos correspondientes al ensayo de identificación realizado sobre la planta real modelada. El ensayo de identificación tiene una duración aproximada de 2.5 horas. En este ensayo, se introducen escalones en la demanda de energía eléctrica, los caudales de los circuitos primario y secundario del sistema de refrigeración, y la demanda de energía térmica. Estas señales se representan en la figura 5.28. Además de este ensayo, se dispone de un ensayo de validación (ver figura 5.29) que no será utilizado para la obtención de los modelos óptimos o casi-óptimos, pero sí para la validación de los mismos.

Se define el MOP como:

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ f_3(\mathbf{x})] \quad (5.30)$$

donde



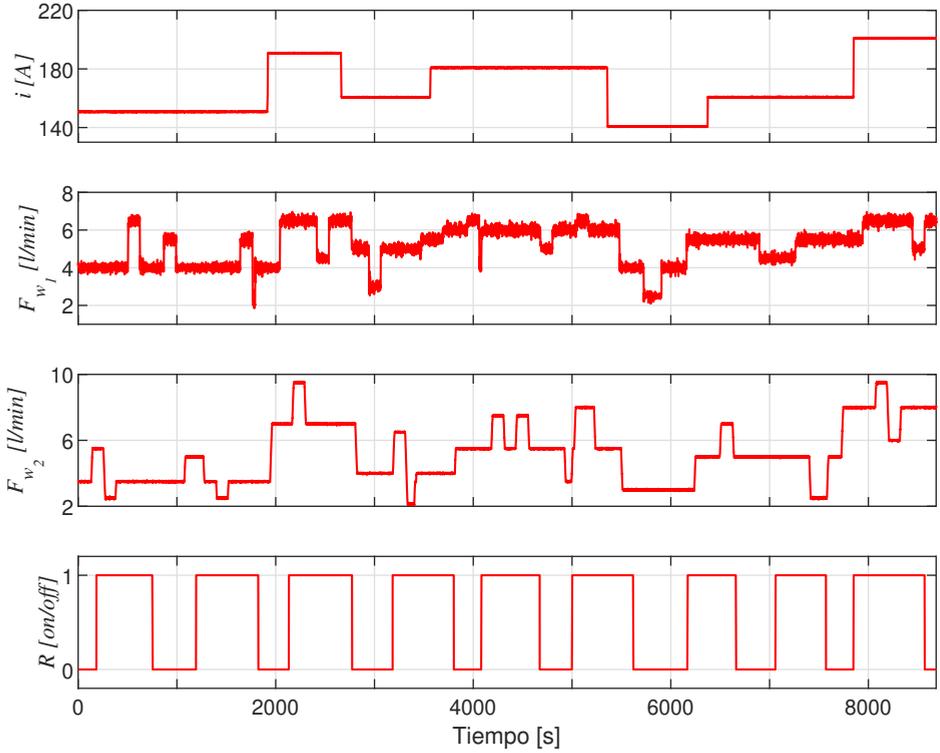
**Figura 5.28:** Ensayo de identificación del proceso de refrigeración de un sistema  $\mu$ -CHP.

$$f_1 = \frac{1}{T} \int_0^T \left| \hat{T}_{w_{out}}(t) - T_{w_{out}}(t) \right| dt \quad (5.31)$$

$$f_2 = \frac{1}{T} \int_0^T \left| \hat{T}_{w_{in}}(t) - T_{w_{in}}(t) \right| dt \quad (5.32)$$

$$f_3 = \frac{1}{T} \int_0^T \left| \hat{T}_{t_2}(t) - T_{t_2}(t) \right| dt \quad (5.33)$$

siendo  $T = 8087$  segundos (duración del ensayo de identificación), las variables con acento circunflejo las salidas del proceso (por ejemplo,  $\hat{T}_{w_{out}}(t)$ ), las variables sin acento circunflejo las salidas del modelo (por ejemplo,  $T_{w_{out}}(t)$ ), y  $\mathbf{x}$  el vector de parámetros:



**Figura 5.29:** Ensayo de validación del proceso de refrigeración de un sistema  $\mu$ -CHP.

$$\mathbf{x} = [V_{t1} \ h_{t_{min}} \ h_{t_{max}} \ h_{s_{min}} \ h_{s_{max}} \ V_{p1} \ h_{p1_{loss}} \ V_w \ V_a \ k_a \ h_{fc2_{max}} \ h_{fc2_{min}} \ h_{fc_{loss}} \ h_{a_{max}} \\ h_{a_{min}} \ h_{w_{max}} \ h_{w_{min}} \ h_{aw} \ cal_{T_{w_{out}}} \ cal_{T_{a_{out}}} \ V_{p4} \ h_{p4_{loss}} \ cal_{T_{p4_{out}}} \ h_{r_{OFFmin}} \\ h_{r_{OFFmax}} \ h_{r_{ONmin}} \ h_{r_{ONmax}} \ V_{t2} \ V_r \ T_{amb_r}]$$

sujeto a:

$$\underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}}$$

siendo  $\underline{\mathbf{x}}$  y  $\bar{\mathbf{x}}$  (ver tabla 5.9) los límites inferiores y superiores del vector de parámetros  $\mathbf{x}$ .

**Tabla 5.9:** Límites inferiores ( $\underline{x}$ ) y superiores ( $\bar{x}$ ) de los parámetros a estimar del sistema de refrigeración.

Parámetro	$\underline{x}$	$\bar{x}$	Parámetro	$\underline{x}$	$\bar{x}$
$V_{t1}$	0.001	0.003	$h_{w_{max}}$	1	200
$h_{t_{min}}$	1	100	$h_{w_{min}}$	1	150
$h_{t_{max}}$	1	200	$h_{aw}$	1	100
$h_{s_{min}}$	1	150	$cal_{T_{w_{out}}}$	-2	2
$h_{s_{max}}$	1	300	$cal_{T_{a_{out}}}$	-2	2
$V_{p1}$	0.0001	0.001	$V_{p4}$	0.001	0.002
$h_{p1_{loss}}$	1	15	$h_{p4_{loss}}$	1	14
$V_w$	0.001	0.004	$cal_{T_{p4_{out}}}$	-2	2
$V_a$	0.001	0.005	$h_{r_{OFFmin}}$	1	40
$k_a$	1000	8000	$h_{r_{OFFmax}}$	1	40
$h_{fc2_{max}}$	1	100	$h_{r_{ONmin}}$	1	100
$h_{fc2_{min}}$	1	100	$h_{r_{ONmax}}$	1	200
$h_{fc_{loss}}$	1	15	$V_{t2}$	0.015	0.035
$h_{a_{max}}$	1	150	$V_r$	0.001	0.005
$h_{a_{min}}$	1	100	$T_{amb_r}$	15	35

La simulación del sistema tiene un paso fijo  $T_s = 0.1$  segundos. Los objetivos de diseño miden el error, a lo largo del ensayo de identificación previamente mencionado, en las temperaturas del agua: de entrada y salida de la pila ( $T_{w_{in}}$  y  $T_{w_{out}}$  respectivamente) y del depósito 2 ( $T_{t2}$ ). Las temperaturas del aire de salida a la pila ( $T_{a_{out}}$ ), y de entrada y salida en la carcasa del intercambiador de calor ( $T_{s_{in}}$  y  $T_{s_{out}}$  respectivamente) son salidas del modelo de menor importancia. Por ello, el error asociado a dichas señales no han sido incluido en los objetivos de diseño. De este modo, se reduce el MOP a 3 objetivos facilitando así el proceso de optimización y el análisis de los modelos obtenidos. Sin embargo, los errores en estas salidas (objetivos  $f_4$ ,  $f_5$  y  $f_6$ , ver ecuación 5.34) serán analizados en la fase de decisión. De manera que ante dos modelos con prestaciones similares, sobre los objetivos de diseño ( $f_1$ ,  $f_2$  y  $f_3$ ), pudiésemos discriminar entre ellos analizando el resto de objetivos ( $f_4$ ,  $f_5$  y  $f_6$ ).

$$\begin{aligned}
 f_4 &= \frac{1}{T} \int_0^T \left| \hat{T}_{a_{out}}(t) - T_{a_{out}}(t) \right| dt \\
 f_5 &= \frac{1}{T} \int_0^T \left| \hat{T}_{s_{out}}(t) - T_{s_{out}}(t) \right| dt \\
 f_6 &= \frac{1}{T} \int_0^T \left| \hat{T}_{s_{in}}(t) - T_{s_{in}}(t) \right| dt
 \end{aligned} \tag{5.34}$$

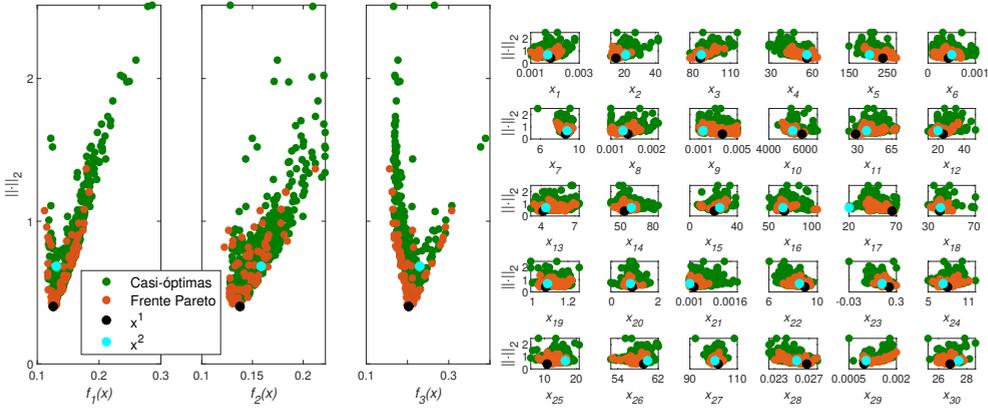
Para optimizar el MOP definido, se utiliza nevMOGA con la siguiente configuración:

- $Nind_G = 4$
- $Nind_P = 250$
- $Iteraciones = 1000$
- $\mathbf{n\_box} = [10 \ 10 \ 10]$
- $\boldsymbol{\epsilon} = [0.01 \ 0.01 \ 0.01]$
- $\mathbf{n} = [0.0005 \ 10 \ 20 \ 10 \ 40 \ 0.0003 \ 4 \ 0.0005 \ 0.0001 \ 1000 \ 15 \ 15 \ 3 \ 15 \ 10 \ 15 \ 15 \ 15 \ 0.5 \ 0.5 \ 0.0005 \ 4 \ 0.2 \ 5 \ 7 \ 7 \ 10 \ 0.002 \ 0.0004 \ 3]$

para la definición de los parámetros restantes, se utilizan los valores sugeridos en [37] para el algoritmo original (ev-MOGA).

En este MOP las variables de decisión y los objetivos tienen un sentido físico. Por ello, la elección de los parámetros de nevMOGA ( $\boldsymbol{\epsilon}$  y  $\mathbf{n}$ ) ha resultado sencilla para el diseñador. La figura 5.30 muestra el conjunto de modelos obtenidos ( $P_{Q,n}$ ) utilizando nevMOGA para el MOP planteado. En primer lugar, se observa cómo existe un gran número de modelos casi-óptimos con rendimiento similar a los óptimos. Para analizar su rendimiento, se eligen dos de ellos. En primer lugar, se selecciona el modelo  $\mathbf{x}^1$ , que es el modelo del frente de Pareto que obtiene un valor inferior de norma 2 (eje de ordenadas de la figura 5.30). Esta alternativa es un modelo compensado, y podría ser la elección final del diseñador. En segundo lugar, elegimos una solución casi-óptima ( $\mathbf{x}^2$ ) dominada por  $\mathbf{x}^1$ , pero en un vecindario diferente, es decir, significativamente diferente.

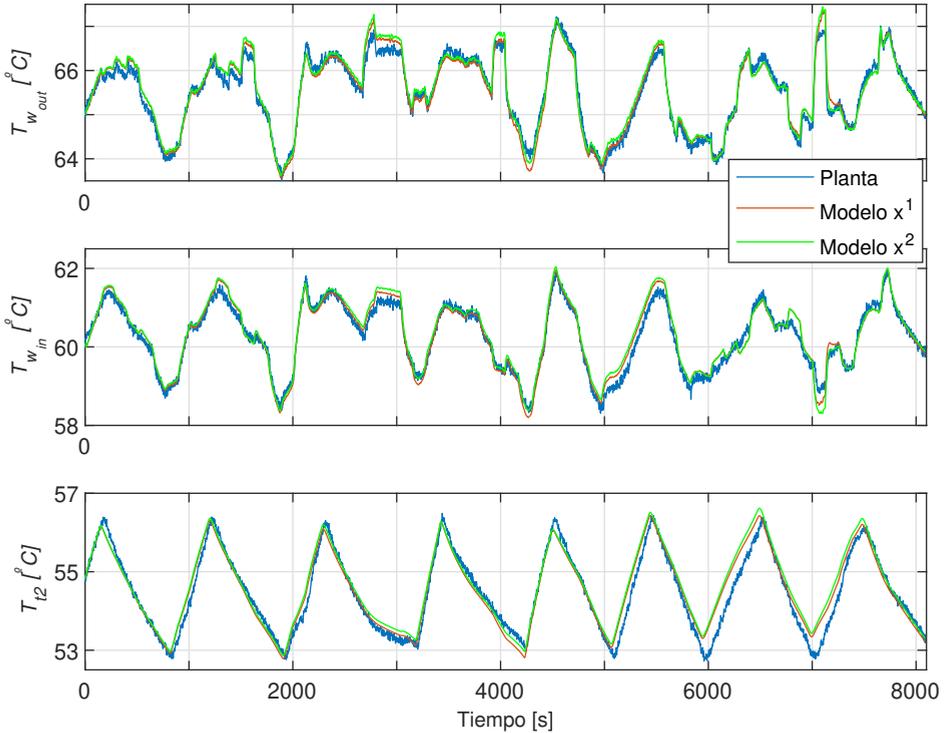
La respuesta de los modelos  $\mathbf{x}^1$  y  $\mathbf{x}^2$  sobre el ensayo de identificación se observa en la figura 5.31. Además, los valores de los objetivos definidos para ambos modelos se observan en la tabla 5.11. El modelo  $\mathbf{x}^1$  obtiene un error ligeramente menor a  $\mathbf{x}^2$ , en todos los objetivos de diseño ( $f_1$  a  $f_3$ ). Sin embargo, esta mejora



**Figura 5.30:** Conjunto de modelos óptimos (naranja) y casi-óptimos no dominados en su vecindad (verde) obtenidos. Los modelos se representan mediante la herramienta de visualización LD, utilizando la norma 2. A la izquierda, se muestra el valor objetivo de cada una de las opciones encontradas. A la derecha, se muestran sus variables de decisión.

es prácticamente inapreciable en su respuesta temporal (ver figura 5.31). Por ello, al ser dos modelos con rendimiento muy similar, creemos que debemos analizar nuevos objetivos para poder tomar una decisión con mayor criterio.

En primer lugar, observamos en la figura 5.32 las salidas de menor importancia ( $T_{a_{out}}$ ,  $T_{s_{in}}$  y  $T_{s_{out}}$ ) de los modelos comparados sobre el ensayo de identificación. En ella, se observa cómo el modelo casi-óptimo  $\mathbf{x}^2$  mejora el rendimiento (error), en todas ellas, al modelo óptimo  $\mathbf{x}^1$  (ver también el valor de los objetivos  $f_4$ ,  $f_5$  y  $f_6$  en la tabla 5.11). Esta mejora es significativa respecto a la salida  $T_{a_{out}}$  (objetivo  $f_4$ ). Sin embargo, pese a que este rendimiento es significativamente mejor, entendiendo como mejor aquel que menor error obtiene, la dinámica del proceso puede no estar bien reflejada por el modelo  $\mathbf{x}^2$ . De hecho, respecto a la dinámica de  $T_{a_{out}}$ , el modelo  $\mathbf{x}^1$  puede ser preferible pese a que tenga errores mayores respecto a la planta real. Los errores obtenidos por el modelo  $\mathbf{x}^1$  son constantes y pueden ser debidos a deficiencias mínimas en el modelo o en el rango de búsqueda de los parámetros definidos. Sin embargo, las diferencias entre las dinámicas del modelo  $\mathbf{x}^2$  y de la planta real, hacen pensar en que este modelo puede tener deficiencias más profundas. Este hecho podría llevar al diseñador a diferentes escenarios si considera relevante estos errores. Por un lado, podría replantearse los objetivos de diseño con el fin de cuantificar en ellos la diferencia entre la dinámica del modelo y de la planta. Los objetivos planteados en el MOP evalúan el error entre el modelo a analizar y la planta real. Estos errores no tienen en cuenta las variaciones dinámicas entre modelo



**Figura 5.31:** Salidas  $T_{w_{out}}$ ,  $T_{w_{in}}$  y  $T_{t2}$  obtenidas para los modelos  $x^1$  y  $x^2$  sobre el ensayo de identificación. El error de estas salidas sobre el ensayo de identificación se evalúa en los objetivos de diseño  $f_1$ ,  $f_2$  y  $f_3$  (ver tabla 5.11).

y planta. Por ello, estas variaciones pueden ser incluidas de algún modo en los objetivos de diseño, planteando un nuevo MOP. Por otro lado, el diseñador podría analizar profundamente posibles deficiencias en el modelo o en el rango de búsqueda de los parámetros, pudiendo evitar así, que determinados modelos obtenidos no representen correctamente la dinámica del proceso real. En este ejemplo, consideramos las diferencias entre las dinámicas del modelo  $x^2$  y la planta, sobre la salida  $T_{a_{out}}$ , despreciables.

En segundo lugar, observamos en la figura 5.33 la respuesta de las seis salidas del modelo sobre el ensayo de validación definido. En la validación se observa cómo los resultados obtenidos por ambos modelos se deterioran. Cierta grado de deterioro de la precisión en la validación es habitual. Sin embargo, creemos que ese deterioro tiene una causa adicional en este caso. La temperatura del aire

**Tabla 5.10:** Modelos  $\mathbf{x}^1$  y  $\mathbf{x}^2$ .

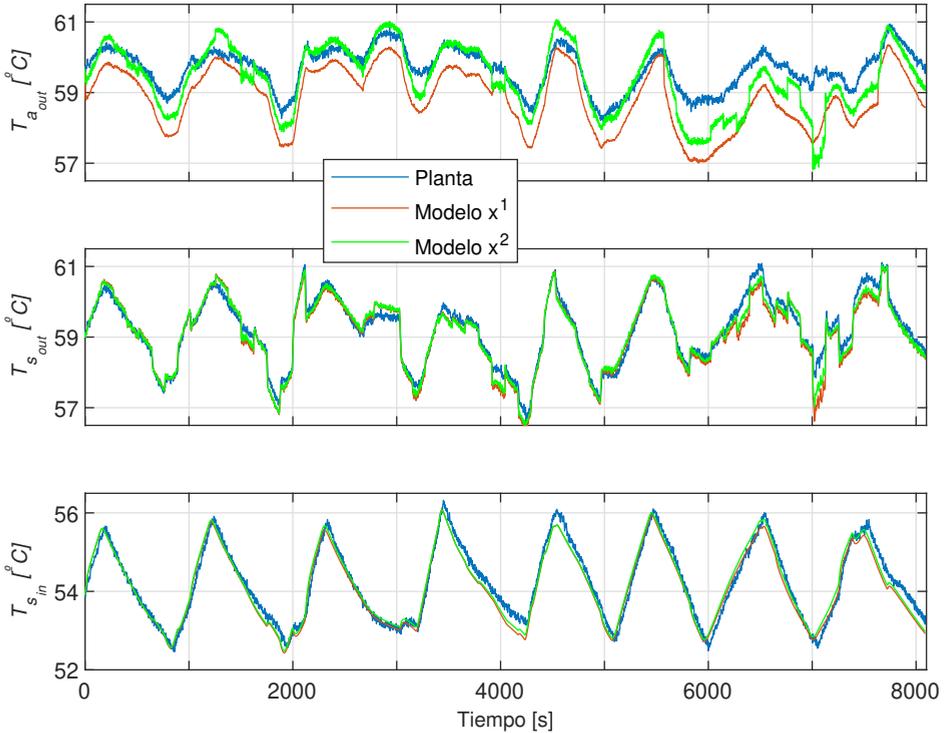
Parámetro	$\mathbf{x}^1$	$\mathbf{x}^2$	Parámetro	$\mathbf{x}^1$	$\mathbf{x}^2$
$V_{t1}$	0.0018	0.0017	$h_{w_{max}}$	68	66.45
$h_{t_{min}}$	14.55	20.78	$h_{w_{min}}$	65.45	20.21
$h_{t_{max}}$	85.8	86.54	$h_{aw}$	39.9	40.53
$h_{s_{min}}$	55.89	55.88	$cal_{T_{w_{out}}}$	1.08	1.09
$h_{s_{max}}$	237.67	202.39	$cal_{T_{a_{out}}}$	0.88	0.8
$V_{p1}$	0.00043	0.0005	$V_{p4}$	0.0011	0.001
$h_{p1_{loss}}$	8.59	8.74	$h_{p4_{loss}}$	8.97	8.52
$V_w$	0.0014	0.0013	$cal_{T_{p4_{out}}}$	0.25	0.2
$V_a$	0.0034	0.0014	$h_{r_{OFFmin}}$	7.8	7.16
$k_a$	5799.3	5295.8	$h_{r_{OFFmax}}$	10.83	16.52
$h_{fc2_{max}}$	27.83	39.05	$h_{r_{ONmin}}$	59.12	59.88
$h_{fc2_{min}}$	23.8	19.86	$h_{r_{ONmax}}$	101.93	100.57
$h_{fc_{loss}}$	4.17	4.41	$V_{t2}$	0.0267	0.0256
$h_{a_{max}}$	53.49	59.89	$V_r$	0.001	0.0011
$h_{a_{min}}$	20.64	25.19	$T_{amb_r}$	26.81	27.42

**Tabla 5.11:** Valor de los objetivos definidos obtenidos para los modelos  $\mathbf{x}^1$  y  $\mathbf{x}^2$ .

Modelo	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$
$\mathbf{x}^1$	<b>0.126</b>	<b>0.138</b>	<b>0.203</b>	0.805	0.200	0.178
$\mathbf{x}^2$	0.131	0.159	0.230	<b>0.405</b>	<b>0.167</b>	<b>0.170</b>

del radiador ( $T_{amb_r}$ ) es un parámetro a ajustar, y por lo tanto, está ajustado teniendo en cuenta los datos de identificación. De este modo, se asume que esta temperatura es idéntica para el ensayo de validación. Esta hipótesis puede ser falsa, y puede ser la causa de un deterioro mayor sobre el ensayo de validación. Asumiendo este deterioro, respecto a los modelo analizados, el modelo casi-óptimo  $\mathbf{x}^2$  mejora el rendimiento, en todas las salidas, al modelo óptimo  $\mathbf{x}^1$  (ver los valores objetivo en la tabla 5.12) sobre el ensayo de validación. Esta mejora es significativa, de nuevo, en la salida  $T_{a_{out}}$  ( $f_4$ ).

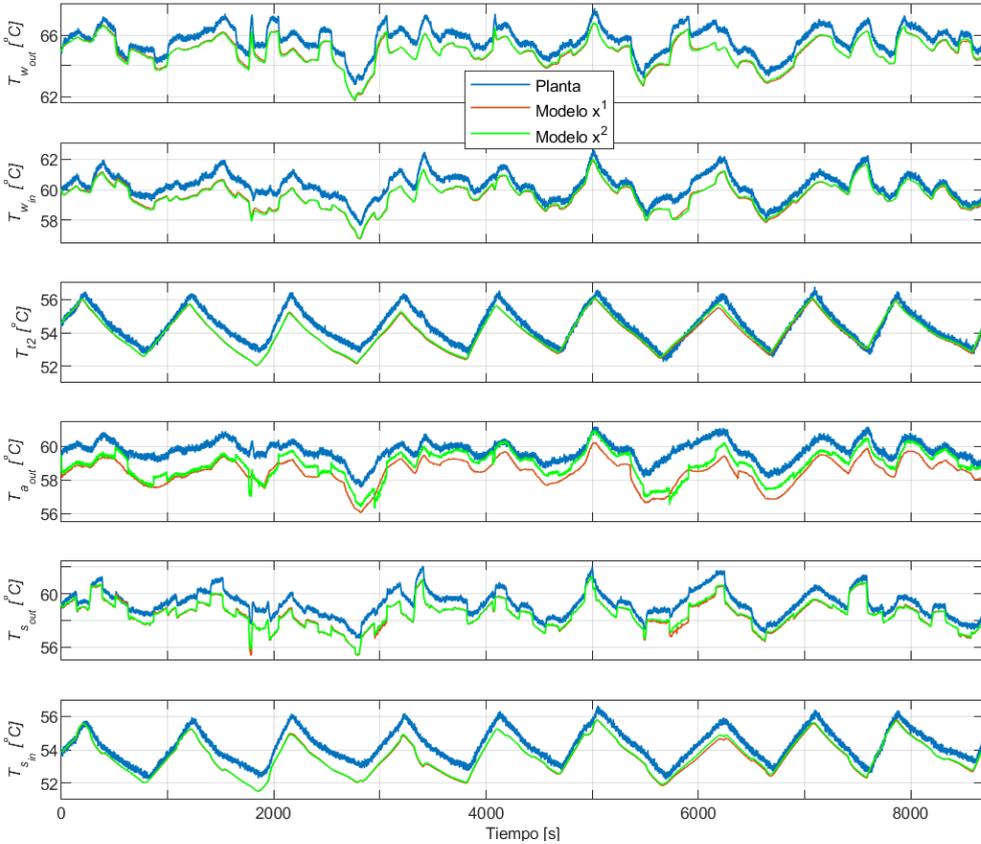
Así pues,  $\mathbf{x}^1$  y  $\mathbf{x}^2$  son dos modelos significativamente diferentes con rendimiento muy similar en los objetivos de diseño. En ellos, el modelo  $\mathbf{x}^1$  obtiene una prestación ligeramente superior. Sin embargo, respecto a otros objetivos (errores relativos a las salidas  $T_{a_{out}}$ ,  $T_{s_{in}}$  y  $T_{s_{out}}$ ), el modelo  $\mathbf{x}^2$  obtiene un



**Figura 5.32:** Salidas  $T_{a_{out}}$ ,  $T_{s_{in}}$  y  $T_{s_{out}}$  obtenidas para los modelos  $x^1$  y  $x^2$  sobre el ensayo de identificación. El error de estas salidas sobre el ensayo de identificación se evalúa en los objetivos  $f_4$ ,  $f_5$  y  $f_6$  (ver tabla 5.11).

rendimiento superior a  $x^1$ , siendo esta mejora significativa respecto a la salida  $T_{a_{out}}$ . Además, respecto al ensayo de validación, el modelo  $x^2$  mejora al modelo óptimo  $x^1$  en todas las salidas, siendo, de nuevo, esta mejora significativa respecto a  $T_{a_{out}}$ . Por todo ello, elegimos el modelo  $x^2$  como la solución final del MOP.

Por lo tanto, gracias a obtener el conjunto de soluciones casi-óptimas no dominadas en su vecindad, el diseñador puede, como se ha demostrado, llevar a cabo un análisis pormenorizado que le ha permitido: 1) obtener modelos con rendimiento similar y con características significativamente diferentes a los óptimos, y 2) obtener información de deficiencias en el modelo diseñado y/o en los objetivos de diseño. De este modo, el diseñador puede llevar a cabo la decisión



**Figura 5.33:** Salidas  $T_{w,out}$ ,  $T_{w,in}$ ,  $T_{t2}$ ,  $T_{a,out}$ ,  $T_{s,in}$  y  $T_{s,out}$  obtenidas para los modelos  $\mathbf{x}^1$  y  $\mathbf{x}^2$  sobre el ensayo de validación. El error de estas salidas sobre el ensayo de validación se evalúa en los objetivos  $f_1$  a  $f_6$  (ver tabla 5.12).

**Tabla 5.12:** Valor de los objetivos definidos, sobre el ensayo de validación, obtenidos para los modelos  $\mathbf{x}^1$  y  $\mathbf{x}^2$ .

Modelo	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$
$\mathbf{x}^1$	0.742	0.639	0.447	1.231	0.816	0.642
$\mathbf{x}^2$	<b>0.726</b>	<b>0.629</b>	<b>0.406</b>	<b>0.827</b>	<b>0.786</b>	<b>0.603</b>

final con una información que no se le hubiera proporcionado con un enfoque multiobjetivo tradicional.

## 5.6 Control del sistema de refrigeración de un sistema $\mu$ -CHP basado en una pila de PEMFC

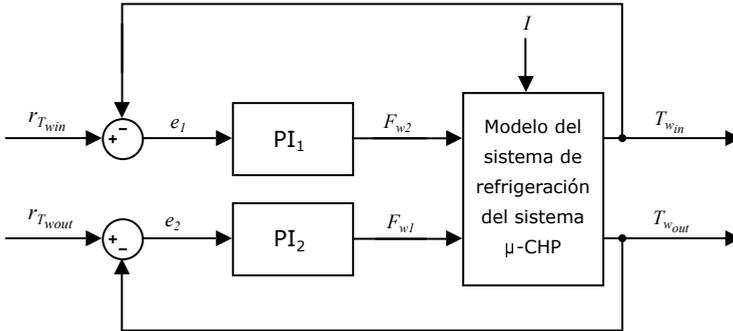
Como se ha comentado en la sección anterior, un sistema  $\mu$ -CHP genera electricidad y calor para el suministro energético de la vivienda. La principal ventaja de estos sistemas es el aprovechamiento de la energía térmica durante el proceso de generación de energía eléctrica. En esta sección, se realiza el diseño de un controlador para el proceso de refrigeración de un sistema  $\mu$ -CHP basado en pila de combustible PEMFC. En estos sistemas, es necesario avanzar en el control de la temperatura de la pila de tipo PEMFC. El correcto diseño del sistema de refrigeración de la pila es vital en la eficiencia de la misma. Por ello, una vez identificado el sistema de refrigeración del sistema  $\mu$ -CHP (ver sección 5.5), en esta sección, se va realizar el diseño del control de las temperaturas del agua de entrada y salida de la pila PEMFC.

La estructura de control está compuesta por dos controladores PI, donde la salida  $T_{w_{out}}$  (temperatura de salida del agua de la pila) se controla mediante la acción de control  $F_{w1}$  (caudal del circuito primario) y  $T_{w_{in}}$  (temperatura de entrada del agua de la pila) mediante  $F_{w2}$  (caudal del circuito secundario). Esto es:

$$\begin{aligned} F_{w2}(s) &= Kc_1 \left( e_1(s) + \frac{1}{Ti_1} \frac{1}{s} e_1(s) \right) \\ F_{w1}(s) &= Kc_2 \left( e_2(s) + \frac{T_s}{Ti_2} \frac{1}{s} e_2(s) \right) \end{aligned} \quad (5.35)$$

donde  $Kc_1$  y  $Kc_2$  son las ganancias proporcionales,  $Ti_1$  y  $Ti_2$  son las constantes de tiempo integrales en segundos, y  $e_1 = r_{T_{w_{in}}} - T_{w_{in}}$  y  $e_2 = r_{T_{w_{out}}} - T_{w_{out}}$  son los errores, donde  $r_{T_{w_{in}}}$  y  $r_{T_{w_{out}}}$  son las referencias para  $T_{w_{in}}$  y  $T_{w_{out}}$  respectivamente (ver figura 5.34).

Para el diseño del controlador de este sistema se mantiene la temperatura del agua del depósito 2 ( $T_{t2}$ ) constante e igual a  $55^\circ C$ . Así, se simplifica el circuito secundario del sistema  $\mu$ -CHP, manteniendo solo la tubería 4 en el Submodelo 3 (ver descripción del modelo en el Apéndice A). Las temperaturas de salidas del sistema real tienen un ruido asociado. En el procesado real, este ruido se filtra con el objetivo de evitar que se propague a las acciones de control del sistema. Del mismo modo, para la simulación del sistema se introduce un ruido

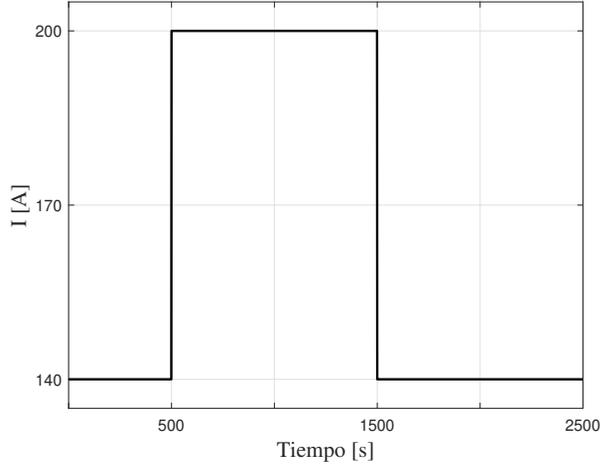


**Figura 5.34:** Esquema de emparejamiento de controladores PI empleado para el control del sistema de refrigeración del sistema  $\mu$ -CHP.

similar al producido en el proceso real. Este ruido también será filtrado con el objetivo de que las acciones de control del sistema se vean menos influenciadas por él. De este modo, se tiene un modelo lo más similar posible al proceso real. La simulación del sistema tiene un paso fijo  $Ts = 0.1$  segundos.

Los objetivos de diseño se evalúan a lo largo de un ensayo definido. Este ensayo tiene dos cambios de la corriente demandada en 500 y 1500 segundos (ver figura 5.35). Además, para que la pila funcione en el punto de operación óptimo, la temperatura de salida del agua de la pila ( $T_{w,out}$ ) debe ser de  $65^{\circ}C$ . Por otro lado, debe existir un gradiente de  $5^{\circ}C$  entre la temperatura del agua de entrada y salida de la pila, tal y como indica el fabricante. Por ello, las referencias del sistema son constantes a lo largo de todo el ensayo:  $r_{T_{w,out}} = 65^{\circ}C$  y  $r_{T_{w,in}} = 60^{\circ}C$ . De este modo, se conseguirá funcionar en el punto de operación óptimo sugerido por el fabricante. Por lo tanto, el rendimiento de los controladores se evalúa ante el rechazo de perturbaciones (demandas de corriente).

Como se ha mencionado en secciones anteriores, agregando los objetivos de diseño conseguimos simplificar los procesos de optimización y decisión. Los controladores obtenidos se evalúan mediante objetivos que miden el rendimiento y el esfuerzo de control. Los objetivos relativos al rendimiento y el esfuerzo de control, presumiblemente, entrarán en conflicto. Por ello, es valioso estudiarlos como objetivos independientes con el fin de analizar la compensación que existe entre cada uno de ellos. Sin embargo, debido a que el sistema consta de dos salidas y dos acciones de control, parece razonable agregar los objetivos relativos al rendimiento (en cada una de las dos salidas, pues tienen misma magnitud y se pueden considerar igual de importantes) y al esfuerzo de control



**Figura 5.35:** Perfil de las demandas de corriente introducidas para el diseño de un controlador del proceso de refrigeración del sistema  $\mu$ -CHP.

(en ambas acciones de control, ya que también tienen mismas unidades e igual relevancia). Por lo tanto, el primer objetivo  $f_1$  se define como la agregación del error medio en ambas salidas ( $T_{w_{out}}$  y  $T_{w_{in}}$ ). El segundo objetivo  $f_2$  se define como la agregación de la integral de la derivada de la acción de control absoluta en ambas acciones de control ( $F_{w1}$  y  $F_{w2}$ ). Por lo tanto, el objetivo  $f_1$  mide el rechazo de perturbaciones del sistema, mientras que  $f_2$  mide el esfuerzo de control. Con ello, se define el MOP como:

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x})] \quad (5.36)$$

donde

$$\begin{aligned} f_1(\mathbf{x}) &= f_{11}(\mathbf{x}) + f_{12}(\mathbf{x}) \\ f_2(\mathbf{x}) &= f_{21}(\mathbf{x}) + f_{22}(\mathbf{x}) \end{aligned} \quad (5.37)$$

donde

$$\begin{aligned}
 f_{11}(\mathbf{x}) &= \frac{\int_{t_i}^{t_f} |e_1| dt}{t_f - t_i} \\
 f_{12}(\mathbf{x}) &= \frac{\int_{t_i}^{t_f} |e_2| dt}{t_f - t_i} \\
 f_{21}(\mathbf{x}) &= \int_{t_i}^{t_f} \left| \frac{dF_{w2}}{dt} \right| dt \\
 f_{22}(\mathbf{x}) &= \int_{t_i}^{t_f} \left| \frac{dF_{w1}}{dt} \right| dt
 \end{aligned} \tag{5.38}$$

sujeto a:

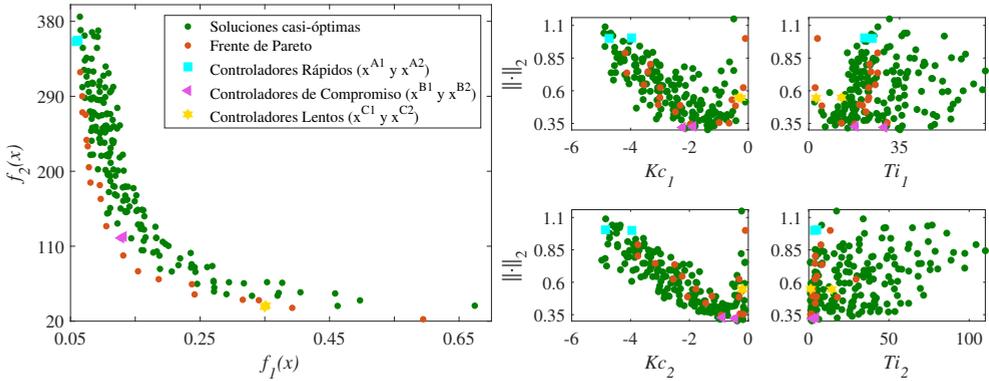
$$\begin{aligned}
 \underline{\mathbf{x}} &\leq \mathbf{x} \leq \bar{\mathbf{x}} \\
 f_1(\mathbf{x}) &< 0.75, f_2(\mathbf{x}) < 400
 \end{aligned}$$

y donde

$$\begin{aligned}
 \mathbf{x} &= [Kc_1, Ti_1, Kc_2, Ti_2] \\
 \underline{\mathbf{x}} &= [-10, 1, -10, 1] \quad \bar{\mathbf{x}} = [-0.1, 250, -0.1, 250] \\
 t_i &= 500 \text{ segundos y } t_f = 2500 \text{ segundos}
 \end{aligned}$$

Una vez definido el problema de optimización, se deben establecer los dos parámetros principales de nevMOGA ( $\epsilon$  y  $\mathbf{n}$ ). En este MOP los objetivos de diseño tienen sentido físico ( $f_1$  mide el error en  $^{\circ}C$  y  $f_2$  mide variaciones en las acciones de control en  $m^3/s$ ). Por ello, resulta sencillo definir  $\epsilon_1 = 0.005^{\circ}C$  y  $\epsilon_2 = 2.5m^3/s$  como las máximas degradaciones aceptables sobre  $f_1$  y  $f_2$  respectivamente ( $\epsilon = [0.005 \ 25]$ ). Una vez elegido  $\epsilon$ , elegimos  $\mathbf{n} = [0.5 \ 10 \ 0.5 \ 10]$  en base al espacio de búsqueda previamente definido (aproximadamente un 4 – 5 % del espacio de búsqueda). Para optimizar el MOP definido, se utiliza nevMOGA con la siguiente configuración:

- $Nind_G = 16$
- $Nind_P = 250$



**Figura 5.36:** Conjunto de controladores óptimos y casi-óptimos no dominados en su vecindad ( $P_{Q,n}$ ) obtenidos para el diseño del sistema de refrigeración. Las variables de decisión se muestran mediante la herramienta de visualización LD, utilizando la norma 2.

- $Iteraciones = 2500$
- $n\_box = [20 \ 20]$

para la definición de los parámetros restantes, se utilizan los valores sugeridos en [37] para el algoritmo original (ev-MOGA).

La figura 5.36 muestra el conjunto de controladores pertenecientes a  $P_{Q,n}$  encontrados por nevMOGA. Para llevar a cabo un análisis más profundo de estos controladores elegimos dos controladores (uno óptimo y uno casi-óptimo) en tres zonas diferentes de  $P_{Q,n}$  (controladores rápidos, de compromiso y lentos). Los controladores  $\mathbf{x}^{A1}$  y  $\mathbf{x}^{A2}$ , son controladores rápidos, es decir, con un rendimiento muy bueno a cambio de unas agresivas acciones de control. Los controladores  $\mathbf{x}^{C1}$  y  $\mathbf{x}^{C2}$ , son controladores lentos, es decir, obtienen un rendimiento peor a cambio de unas suaves acciones de control. Por último, los controladores  $\mathbf{x}^{B1}$  y  $\mathbf{x}^{B2}$  son controladores de compromiso. Para mostrar las variables de decisión, empleamos la herramienta de visualización LD, utilizando la norma 2. Con lo que respecta a los objetivos de diseño, al tener solo dos, es preferible analizar su compensación en una sola gráfica. Por este motivo, no hemos empleado LD para la representación de los valores objetivos de los controladores. Como se observa en la figura, nevMOGA ha sido capaz de encontrar un gran número de controladores casi-óptimos, no dominados en su vecindad, con prestaciones similares a los óptimos. Estos controladores proporcionan al diseñador una información valiosa para tomar una decisión final con mayor criterio.

**Tabla 5.13:** Controladores rápidos ( $x^{A1}$  y  $x^{A2}$ ), de compromiso ( $x^{B1}$  y  $x^{B2}$ ) y lentos ( $x^{C1}$  y  $x^{C2}$ ).

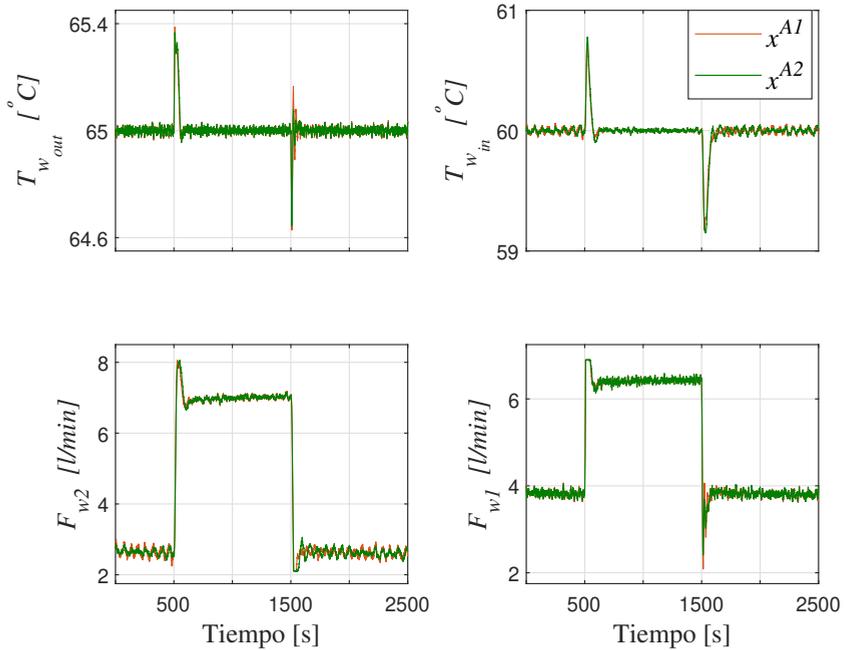
Controlador	$K_{c1}$	$T_{i1}$	$K_{c2}$	$T_{i2}$
$x^{A1}$	-4.73	24.47	-3.95	3.55
$x^{A2}$	-3.97	21.51	-4.85	4.96
$x^{B1}$	-2.23	28.98	-0.46	2.18
$x^{B2}$	-1.87	17.87	-0.88	4.45
$x^{C1}$	-0.24	2.86	-0.21	14.31
$x^{C2}$	-0.33	12.67	-0.19	1.23

**Tabla 5.14:** Valor objetivo de los controladores rápidos, de compromiso y lentos.

Controlador	$f_{11}(x)$	$f_{12}(x)$	$f_1(x)$	$f_{21}(x)$	$f_{22}(x)$	$f_2(x)$
$x^{A1}$	<b>0.0421</b>	0.0168	<b>0.0589</b>	190.5	<b>165.3</b>	<b>355.8</b>
$x^{A2}$	0.0458	<b>0.0158</b>	0.0616	<b>161.2</b>	196.1	357.4
$x^{B1}$	<b>0.083</b>	0.045	<b>0.1280</b>	93.4	<b>26.5</b>	<b>119.9</b>
$x^{B2}$	0.0891	<b>0.0427</b>	0.132	<b>80.5</b>	41.1	121.6
$x^{C1}$	<b>0.138</b>	0.211	<b>0.350</b>	24.7	<b>12.4</b>	<b>37.1</b>
$x^{C2}$	0.251	<b>0.101</b>	0.352	<b>21.1</b>	17.7	38.8

Ahora, fijémonos en los controladores rápidos ( $x^{A1}$  y  $x^{A2}$ , ver tabla 5.13). La figura 5.37 muestra el comportamiento del sistema con ambos controladores. Los valores objetivo de ambos controladores se observan en la tabla 5.14. El error en  $T_{w_{in}}$  ( $f_{11}$ ) es mayor para el controlador casi-óptimo  $x^{A2}$ . Sin embargo, en la salida  $T_{w_{out}}$  ocurre lo contrario ( $f_{12}$ ), el controlador óptimo  $x^{A1}$  obtiene un error mayor. Con lo que respecta al esfuerzo de control, el controlador  $x^{A1}$  obtiene acciones de control más suaves para  $F_{w1}$  ( $f_{22}$ ), y más bruscas para  $F_{w2}$  ( $f_{21}$ ), que el controlador  $x^{A2}$  (ver tabla 5.14). Por lo tanto, pese a que  $x^{A1}$  domina ligeramente a  $x^{A2}$ , no se observa una mejora significativa (sino pequeñas diferencias de comportamiento), lo cual nos hace buscar nuevos indicadores para llevar a cabo la decisión final.

Para estudiar los controladores seleccionados, vamos a llevar a cabo un análisis en 4 escenarios diferentes:



**Figura 5.37:** Respuesta de los controladores  $x^{A1}$  (óptimo) y  $x^{A2}$  (casi-óptimo). Aunque  $f(x^{A1}) \preceq f(x^{A2})$ , no se aprecia una clara ventaja del controlador  $x^{A1}$  sobre  $x^{A2}$ .

- Aumento del paso fijo de simulación (inicialmente  $Ts = 0.1$  segundos)
- Degradación eléctrica de la pila de combustible.
- Cambios en el ruido introducido en las salidas del sistema (cambio en la semilla, es decir, en la secuencia de valores del ruido, o en la amplitud del mismo).
- Incertidumbre en el modelo.

La elección del paso fijo de simulación puede variar el conjunto de controladores óptimos (frente de Pareto). Por ello, en el estudio de los controladores seleccionados vamos a analizar como afecta el aumento de este periodo al rendimiento de los controladores seleccionados (mediante los objetivos de diseño). De este modo, con lo que respecta a los controladores rápidos, si aumentamos el paso fijo de simulación a  $Ts = 0.4$  segundos la dominancia de  $x^{A1}$  sobre

**Tabla 5.15:** Valores objetivo de los controladores  $\mathbf{x}^{A1}$  y  $\mathbf{x}^{A2}$  ante diferentes escenarios.

Contexto	Controlador	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$
$T_s = 0.4$ segundos	$\mathbf{x}^{A1}$	0.0654	817.0
	$\mathbf{x}^{A2}$	<b>0.0613</b>	<b>792.4</b>
Degradación de la pila del 6 %	$\mathbf{x}^{A1}$	<b>0.0614</b>	349.3
	$\mathbf{x}^{A2}$	0.0637	<b>348.1</b>
Cambio aleatorio del ruido	$\mathbf{x}^{A1}$	0.0651	359.8
	$\mathbf{x}^{A2}$	<b>0.063</b>	<b>359.1</b>

$\mathbf{x}^{A2}$  se invierte, es decir, el controlador casi-óptimo  $\mathbf{x}^{A2}$  domina al controlador óptimo  $\mathbf{x}^{A1}$ . Los valores objetivos en este contexto se observan en la tabla 5.15.

La potencia eléctrica proporcionada por la pila, depende también de su degradación. El fabricante proporciona una curva características Voltaje-Corriente para determinadas temperaturas de funcionamiento de la pila. Estas curvas son válidas al inicio de la vida útil de la pila. Sin embargo, con el paso de las horas de funcionamiento, la potencia eléctrica generada por la pila disminuye. La curva característica proporcionada se degrada, proporcionando menor tensión con la misma corriente demandada, lo que se traduce en una menor energía eléctrica [87] [98]. Por ejemplo, en [98] se delimita esta pérdida en un 0.3 – 3% de pérdida en el voltaje proporcionado por la pila por cada 1000 horas de funcionamiento. Por ello, en este análisis, se va a estudiar cómo afecta esta degradación de la pila en los controladores  $\mathbf{x}^{A1}$  y  $\mathbf{x}^{A2}$ . Este estudio se realiza reduciendo en un porcentaje (degradación) la tensión proporcionada por la pila, respecto a la tensión proporcionada por el fabricante al inicio de su vida útil (la que se considera en el modelo inicialmente). Fijándonos en los controladores rápidos, considerando una degradación de la pila del 6%, la dominancia de  $\mathbf{x}^{A1}$  sobre  $\mathbf{x}^{A2}$  deja de existir (ver tabla 5.15). En este caso, el controlador  $\mathbf{x}^{A2}$  proporciona acciones de control más suaves. Por lo tanto, observamos como con el paso del tiempo, el controlador casi-óptimo  $\mathbf{x}^{A2}$  deja de ser peor (entendiendo como peor aquel que mayor valor objetivo obtiene) que el controlador óptimo  $\mathbf{x}^{A1}$ .

Las temperaturas de salidas del sistema real tienen un ruido asociado. Este ruido se filtra con el objetivo de evitar que se propague a las acciones de control del sistema. Este ruido ha sido incluido en el sistema simulado de forma análoga, mediante un ruido blanco. Sin embargo, el ruido introducido,

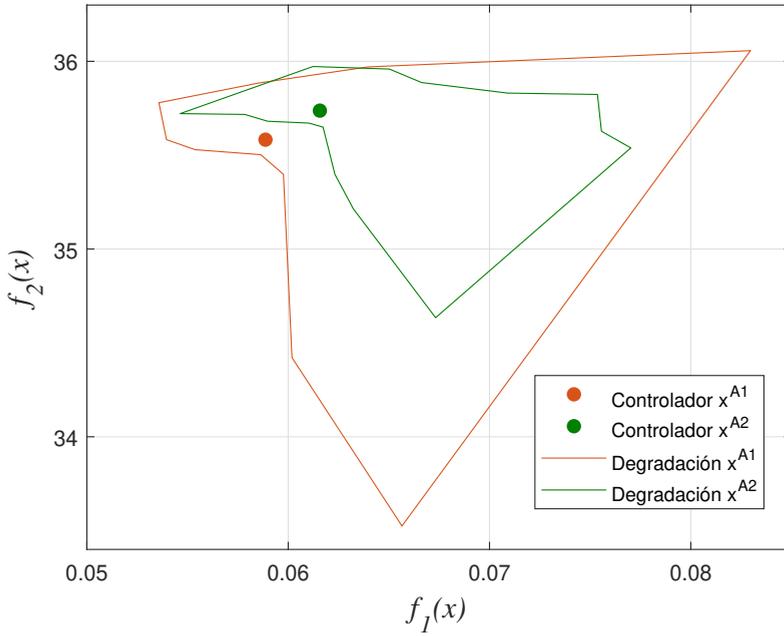
tiene una componente aleatoria en función de su semilla. Este valor modifica la secuencia de los valores del ruido en cada instante, manteniendo la naturaleza del mismo. Por ello, en este análisis, se va a estudiar cómo afecta la componente aleatoria del ruido (cambiando su semilla). De este modo, en esta comparación, modificando la semilla, la dominancia del controlador  $\mathbf{x}^{A1}$  sobre  $\mathbf{x}^{A2}$  se invierte ( $\mathbf{x}^{A2}$  domina a  $\mathbf{x}^{A1}$ , ver tabla 5.15). Esto implica, que en otro contexto (igual de válido que el escenario inicial, para el diseñador), el controlador  $\mathbf{x}^{A2}$  podría ser óptimo (y  $\mathbf{x}^{A1}$  casi-óptimo), y por ello, ambos controladores son igual de buenos.

Por último, se va a analizar cómo afecta la incertidumbre del modelo en los controladores estudiados. Para ello, realizamos 50 variaciones de los parámetros del modelo con una variación de  $\pm 15\%$ . Los controladores se evaluarán, mediante los objetivos de diseño, sobre cada una de las 50 variaciones del modelo. De este modo, medimos la robustez de los controladores. La figura 5.38 muestra los límites de la degradación de los controladores  $\mathbf{x}^{A1}$  y  $\mathbf{x}^{A2}$  sobre las 50 variaciones de la planta. Como se observa, la degradación de  $\mathbf{x}^{A2}$  está prácticamente incluida dentro de los límites de degradación de  $\mathbf{x}^{A1}$ . Por lo tanto, se puede concluir que el controlador  $\mathbf{x}^{A2}$  es más robusto, teniendo menor variabilidad ante incertidumbre en los parámetros del modelo.

Por lo tanto, después de llevar a cabo este estudio pormenorizado de los controladores rápidos ( $\mathbf{x}^{A1}$  y  $\mathbf{x}^{A2}$ ),  $\mathbf{x}^{A2}$  puede ser preferido por el diseñador. Ante diferentes contextos (paso fijo de simulación mayor, degradación de la pila, cambio aleatorio del ruido, análisis de robustez) este controlador ha obtenido mejor rendimiento (o igual) a  $\mathbf{x}^{A1}$ . Por lo tanto, en este caso, considerar el controlador casi-óptimo  $\mathbf{x}^{A2}$  ha sido de gran utilidad para el diseñador. No considerarlo (obteniendo solo los controladores óptimos), sería obviar una información relevante que puede variar la decisión final.

Ahora, fijémonos en los controladores de compromiso ( $\mathbf{x}^{B1}$  y  $\mathbf{x}^{B2}$  en la figura 5.36). La figura 5.39 muestra las respuestas del sistema de ambos controladores. Los controladores y sus valores objetivo se observan en las figuras 5.13 y 5.14 respectivamente. El controlador óptimo  $\mathbf{x}^{B1}$  obtiene un error menor sobre  $T_{w_{in}}$  ( $f_{11}$ ) con acciones de control más suaves sobre  $F_{w1}$  ( $f_{22}$ ). Sin embargo, el controlador casi-óptimo  $\mathbf{x}^{B2}$  obtiene menor error sobre  $T_{w_{out}}$  ( $f_{12}$ ) con acciones de control más suaves sobre  $F_{w2}$  ( $f_{21}$ ). De nuevo, pese a que  $\mathbf{x}^{B1}$  domina ligeramente a  $\mathbf{x}^{B2}$ , no se observa una mejora significativa, lo cual nos hace buscar nuevos indicadores para llevar a cabo la decisión final con mayor criterio.

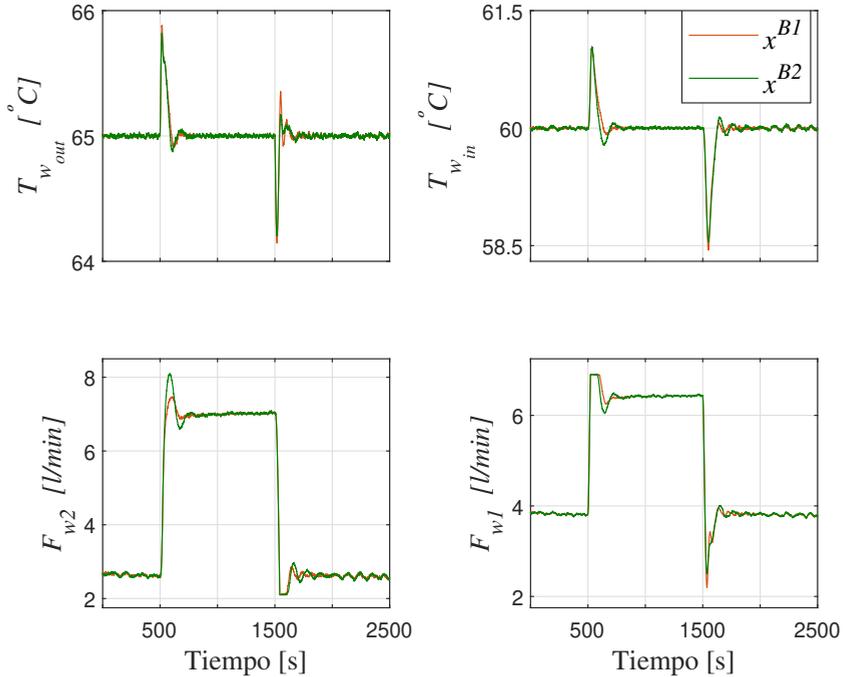
En primer lugar, realizamos el estudio del efecto del aumento del paso fijo de simulación, de forma análoga al estudio de los controladores rápidos realizada



**Figura 5.38:** Límites de degradación de los controladores  $\mathbf{x}^{A1}$  (óptimo) y  $\mathbf{x}^{A2}$  (casi-óptimo) ante la incertidumbre en el modelo. Variando en  $\pm 15\%$  los parámetros del modelo, el rendimiento de los controladores  $\mathbf{x}^{A1}$  y  $\mathbf{x}^{A2}$  se encuentran dentro de los límites mostrados.

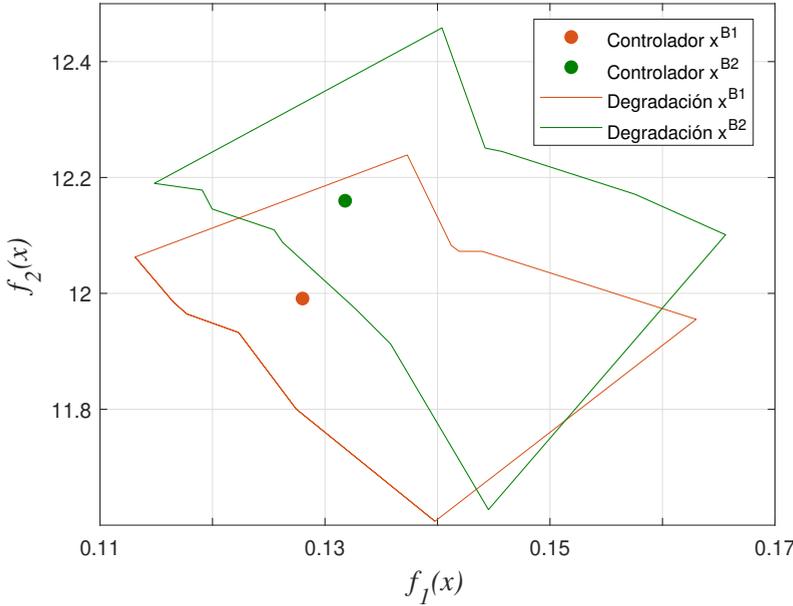
**Tabla 5.16:** Valores objetivo de los controladores  $\mathbf{x}^{B1}$  y  $\mathbf{x}^{B2}$  ante diferentes contextos.

Contexto	Controlador	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$
$T_s = 0.4$ segundos	$\mathbf{x}^{B1}$	<b>0.127</b>	<b>291.4</b>
	$\mathbf{x}^{B2}$	0.134	296.7
Degradación de la pila del 8%	$\mathbf{x}^{B1}$	0.152	112.7
	$\mathbf{x}^{B2}$	<b>0.152</b>	<b>109.2</b>
Disminuir ruido sobre $T_{w_{out}}$	$\mathbf{x}^{B1}$	<b>0.127</b>	117.7
	$\mathbf{x}^{B2}$	0.131	<b>117.3</b>



**Figura 5.39:** Respuesta de los controladores  $\mathbf{x}^{B1}$  (óptimo) y  $\mathbf{x}^{B2}$  (casi-óptimo). Aunque  $f(\mathbf{x}^{B1}) \preceq f(\mathbf{x}^{B2})$ , no se aprecia una clara ventaja del controlador  $\mathbf{x}^{B1}$  sobre  $\mathbf{x}^{B2}$ .

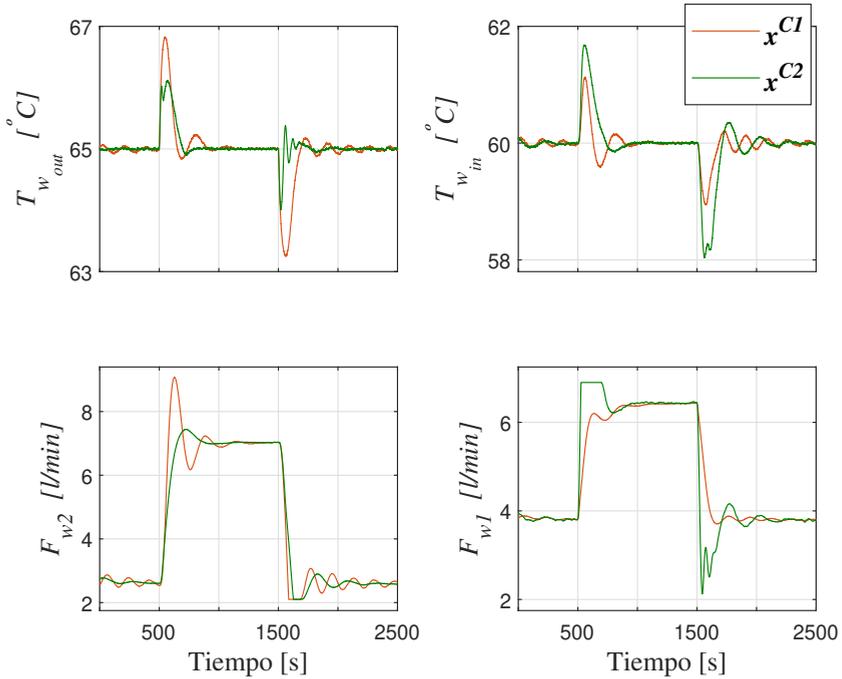
anteriormente. En este caso, la dominancia del controlador  $\mathbf{x}^{B1}$  sigue existiendo pese al aumento del paso fijo de simulación (ver tabla 5.16). En segundo lugar, analizamos cómo afecta la degradación de la pila a los controladores estudiados. Ante una degradación de la pila del 8% (o superior) la dominancia de  $\mathbf{x}^{B1}$  sobre  $\mathbf{x}^{B2}$  se invierte (ver tabla 5.16). Esto implica que con el paso del tiempo, el controlador  $\mathbf{x}^{B2}$  obtendrá un mejor rendimiento (respecto a los objetivos de diseño) que el controlador óptimo  $\mathbf{x}^{B1}$ . En tercer lugar, analizamos la influencia del ruido introducido en las salidas del sistema en los controladores estudiados. En este análisis, se observa cómo al disminuir ligeramente la amplitud del ruido introducido sobre  $T_{w_{out}}$  la dominancia de  $\mathbf{x}^{B1}$  deja de existir (ver tabla 5.16). De igual modo, si aumentamos ligeramente la amplitud del ruido introducido sobre  $T_{w_{in}}$ , de nuevo, el controlador  $\mathbf{x}^{B2}$  ya no está dominado por  $\mathbf{x}^{B1}$ . Finalmente, con lo que respecta al análisis de robustez (realizado de igual modo que en la comparación anterior), se observa que  $\mathbf{x}^{B1}$  parece un controlador más robusto que  $\mathbf{x}^{B2}$  (ver figura 5.40). Por lo tanto, después de



**Figura 5.40:** Límites de degradación de los controladores  $x^{B1}$  (óptimo) y  $x^{B2}$  (casi-óptimo) ante incertidumbre en el modelo variando en  $\pm 15\%$  los parámetros del modelo.

llevar a cabo este estudio pormenorizado de los controladores de compromiso ( $x^{B1}$  y  $x^{B2}$ ), podemos concluir que  $x^{B2}$ , en determinados contextos, es igual de bueno que el controlador óptimo  $x^{B1}$ . Sin embargo, en este caso no está clara la preferencia del controlador casi-óptimo  $x^{B2}$  sobre  $x^{B1}$ . De nuevo, esta información es muy valiosa para el diseñador antes de tomar la decisión final.

Ahora, fijémonos en los controladores lentos ( $x^{C1}$  y  $x^{C2}$ , ver figura 5.36). Los valores objetivo de ambos controladores se observan en la tabla 5.14. La figura 5.41 muestra las salidas y acciones de control de ambos controladores. Ambos controladores tienen un comportamiento significativamente diferente pese a tener un valor objetivo similar. Por ello, parece razonable que sea el diseñador el que elija entre ellos, y no el optimizador el que descarte uno de ellos por tener un valor objetivo ligeramente mejor, en el escenario concreto definido para el diseño del MOP (en otro contexto igual de válido puede no ser mejor). El error en  $T_{w_{in}}$  es mayor para el controlador casi-óptimo  $x^{C2}$  ( $f_{11}$ ). Sin embargo, para la salida  $T_{w_{out}}$  ocurre lo contrario, el controlador óptimo  $x^{C1}$  obtiene un error mayor ( $f_{12}$ ). Con lo que respecta al esfuerzo de control, el controlador  $x^{C1}$  obtiene acciones de control más suaves para  $F_{w1}$  ( $f_{22}$ ) y más bruscas para



**Figura 5.41:** Respuesta de los controladores  $x^{C1}$  (óptimo) y  $x^{C2}$  (casi-óptimo). Ambas respuestas son significativamente diferentes, aunque tiene un rendimiento similar.

$F_{w2}$  ( $f_{21}$ ) que el controlador  $x^{C2}$  (ver tabla 5.14). Por lo tanto,  $x^{C1}$  domina ligeramente a  $x^{C2}$ , pero sus diferencias significativas en su comportamiento nos invitan a realizar un estudio más profundo de ambos controladores con el objetivo de tomar una decisión más informada.

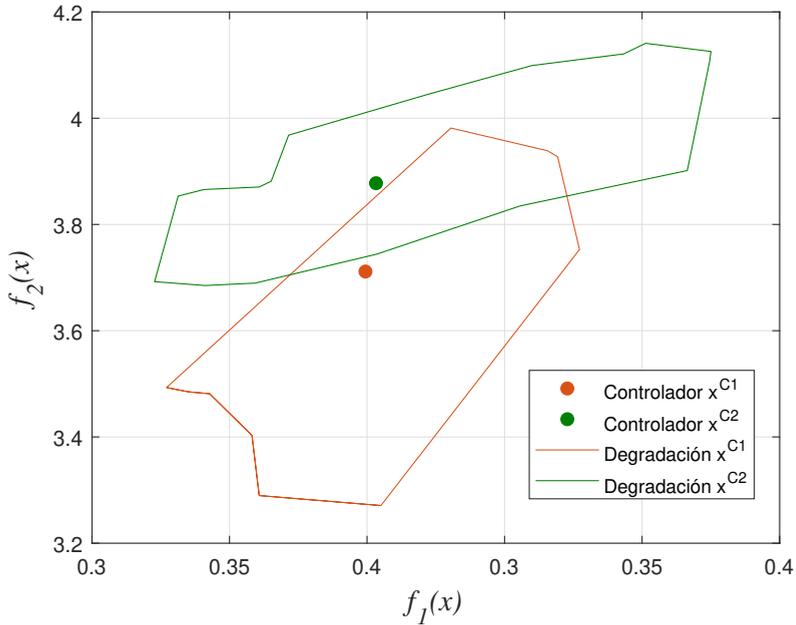
En primer lugar, realizamos el estudio del efecto del aumento del paso fijo de simulación, de forma análoga al estudio de los controladores anteriores. En este caso, la dominancia del controlador  $x^{C1}$  deja de existir con un paso fijo  $T_s = 0.4$  segundos (ver tabla 5.17). En segundo lugar, analizamos cómo afecta la degradación de la pila a los controladores estudiados. En este caso, la dominancia del controlador  $x^{C1}$  no se ve afectada por la degradación de la pila (ver tabla 5.17). En tercer lugar, si modificamos la semilla del ruido (cambio de la secuencia de valores del ruido sin modificar su naturaleza), la dominancia del controlador  $x^{C1}$  deja de existir (ver tabla 5.17). Esto implica que en otro contexto,  $x^{C2}$  podría ser un controlador óptimo. Finalmente, con lo que respecta al análisis de robustez (realizado de igual modo que en las comparaciones an-

**Tabla 5.17:** Valores objetivo de los controladores  $\mathbf{x}^{C1}$  y  $\mathbf{x}^{C2}$  ante diferentes escenarios.

Contexto	Controlador	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$
$T_s = 0.4$ segundos	$\mathbf{x}^{C1}$	0.351	<b>125.1</b>
	$\mathbf{x}^{C2}$	<b>0.351</b>	127.1
Degradación de la pila del 2 %	$\mathbf{x}^{C1}$	<b>0.351</b>	<b>37.9</b>
	$\mathbf{x}^{C2}$	0.363	39.2
Cambio aleatorio del ruido	$\mathbf{x}^{C1}$	0.354	<b>37.3</b>
	$\mathbf{x}^{C2}$	<b>0.352</b>	38.8

teriores), se observa que  $\mathbf{x}^{C1}$  parece un controlador más robusto que  $\mathbf{x}^{C2}$  (ver figura 5.42). Por lo tanto, después de llevar a cabo este estudio pormenorizado de los controladores lentos ( $\mathbf{x}^{C1}$  y  $\mathbf{x}^{C2}$ ), el controlador casi-óptimo  $\mathbf{x}^{C2}$  podría ser óptimo en otro contexto. De hecho, en determinados escenarios, que no son menos preferidos para el diseñador (como un ruido con una semilla diferente), ambos controladores comparados pueden considerarse igual de buenos. De nuevo, esta información es muy útil para el diseñador antes de tomar la decisión final.

Así pues, en esta sección se ha llevado a cabo un estudio para el diseño del control de las temperaturas del agua de entrada y salida de la pila PEMFC del sistema  $\mu$ -CHP. Mediante la metodología empleada, se ha observado la utilidad de considerar los controladores casi-óptimos que podrían ser iguales (o mejores) que los óptimos en diferentes escenarios (paso fijo de simulación diferente, degradación de la pila, cambio del ruido y análisis robustez). Analizar todos estos contextos en los objetivos de diseño es inabordable. Por ello, es útil para el diseñador llevar a cabo este análisis pormenorizado una vez obtenido el conjunto  $P_{Q,n}$ . Gracias a obtener este conjunto, se han obtenido controladores casi-óptimos significativamente diferentes a los óptimos, y que por lo tanto, son relevantes para el diseñador de cara a tomar una decisión final más informada.



**Figura 5.42:** Límites de degradación de los controladores  $x^{C1}$  (óptimo) y  $x^{C2}$  (casi-óptimo) ante incertidumbre en el modelo variando en  $\pm 15\%$  los parámetros del modelo.

## 5.7 Conclusiones

En este capítulo se ha empleado la nueva metodología de optimización multiobjetivo, presentada en la tesis, sobre problemas de control y modelado de sistemas multivariables. Esta nueva metodología considera además del frente de Pareto, el conjunto de alternativas potencialmente útiles, es decir, las soluciones casi-óptimas no dominadas en su vecindad. Como se ha mencionado en capítulos anteriores, estas soluciones proporcionan al diseñador diversidad en el conjunto de soluciones encontradas, sin aumentar en exceso el número de alternativas posibles.

Cuando se tratan problemas multivariables suelen existir diversos objetivos a optimizar. Cuando esto ocurre, las etapas de optimización y decisión se vuelven más complejas. Por ello, es común agregar ciertos objetivos con el fin de simplificar ambas etapas. Sin embargo, agregar los objetivos tiene un inconveniente: alternativas óptimas pasan a ser casi-óptimas. Las alternativas casi-óptimas son ignoradas en un MOP tradicional, y por ello, se pierde mucha informa-

ción al agregar objetivos utilizando el enfoque de optimización multiobjetivo tradicional. Sin embargo, nevMOGA es capaz de obtener las alternativas casi-óptimas no dominadas en su vecindad. De este modo, el diseñador es capaz de recuperar gran parte de esta información perdida. Así pues, agregando objetivos y utilizando el algoritmo nevMOGA, conseguimos simplificar las etapas de optimización y decisión sin perder la información potencialmente útil para el diseñador.

En un MOP, es posible disponer de objetivos que no son incluidos en la fase de optimización. Esto puede deberse a: decisión del diseñador, limitación de recursos o ayuda a la decisión. Un ejemplo de objetivos no incluidos en fase de optimización debido a la decisión del diseñador son los objetivos de validación.

En la sección 5.5, se introduce el diseño de un modelo de refrigeración de un sistema  $\mu$ -CHP. En este diseño se dispone inicialmente de dos ensayos: uno de identificación y uno de validación. El diseñador opta por no incluir la información relativa al ensayo de validación en la fase de optimización. Sin embargo, si desea incluir esta información en la fase de decisión, para seleccionar el modelo final. Con ello, se observa cómo existen modelos casi-óptimas que, teniendo un comportamiento muy similar sobre el ensayo de identificación, obtienen una mejora significativa sobre el ensayo de validación. Este análisis permite al diseñador tomar una decisión más informada.

En la sección 5.2, se introduce el diseño de un control multivariable. En este diseño una vez calculado el conjunto de controladores óptimas y casi-óptimos ( $P_{Q,n}$ ), se define un nuevo indicador para medir la robustez de los controladores obtenidos. Este nuevo indicador tiene un coste computacional alto (50 veces superior al de los objetivos de diseño), lo que provoca que no haya sido incluidos en la fase de optimización debido a una limitación de recursos. Sin embargo, analizando dicho indicador sobre el conjunto  $P_{Q,n}$  obtenido, se observa cómo soluciones casi-óptimas obtienen una robustez mayor a soluciones del frente de Pareto. De nuevo, esta nueva información permite al diseñador estar en disposición de poder tomar una mejor decisión final.

En el *benchmark* de control CIC2018 (ver sección 5.4) se propone el diseño de un control multivariable mediante los datos de un escenario inicial (proporcionado por el concurso). Una vez ya generado  $P_{Q,n}$ , el concurso proporciona un nuevo escenario de funcionamiento del que no se disponía información inicialmente. De este modo, la información relativa al funcionamiento de los controladores obtenidos sobre este nuevo escenario, que no está incluida en la obtención del conjunto  $P_{Q,n}$ , ayudará al diseñador a seleccionar el controlador final. En este análisis, se observa cómo las soluciones del frente de Pareto ob-

tienen una respuesta indeseada en este nuevo escenario. Gracias a encontrar el conjunto  $P_{Q,n}$ , se observa cómo soluciones casi-óptimas sí obtienen un adecuado comportamiento en este nuevo escenario. Por ello, el diseñador decanta su elección hacia controladores casi-óptimos que, obteniendo un comportamiento similar en el escenario inicial, obtienen un comportamiento significativamente mejor en el nuevo escenario planteado por el concurso.

Por último, en el diseño de un controlador para las temperaturas del agua de entrada y salida de la pila PEMFC del sistema  $\mu$ -CHP (ver sección 5.6), se realiza un estudio pormenorizado de los controladores obtenidos. Este análisis se realiza en diversos escenarios (paso fijo de simulación mayor, degradación de la pila, cambio de la semilla del ruido, análisis de robustez). Por ejemplo, se observa cómo un cambio en el paso fijo de simulación o en la degradación de la pila puede variar el conjunto de controladores óptimos, llevando a controladores casi-óptimos al frente de Pareto. Cabe destacar el estudio ente cambios en la semilla del ruido (cambio de la secuencia de valores del ruido, pero no de la naturaleza del mismo) introducido sobre las salidas. Este valor normalmente es elegido de forma aleatoria o por defecto en el entorno de simulación empleado. En este análisis, se observa cómo un simple cambio en la semilla del ruido puede provocar que soluciones casi-óptimas se conviertan en óptimas y viceversa. Este es el caso más claro de la utilidad de las soluciones casi-óptimas, ya que en un contexto diferente, pero exactamente igual de válido para el diseñador, estas soluciones pueden ser iguales (o mejores) a las óptimas. Además se ha realizado un análisis de robustez de los controladores. Este estudio es computacionalmente elevado, y por ello no ha sido posible incluirlo en los objetivos de diseño. Sin embargo, se observa cómo un controlador casi-óptimo obtiene una robustez mejor a uno óptimo. Por lo tanto, en este capítulo se ha puesto de manifiesto la utilidad de considerar el conjunto de alternativas casi-óptimas no dominadas en su vecindad. Gracias a ellas, el diseñador obtiene una mayor información, que le permite llevar a cabo la decisión final con mayor criterio, lo que puede suponer una mejor decisión final.



## Capítulo 6

# Conclusiones y trabajos futuros



## 6.1 Conclusiones

La principal aportación de la tesis recae en la presentación de una nueva metodología de optimización multiobjetivo, destinada al ajuste óptimo de parámetros de modelos y controladores multivariables. Dicha metodología tiene en cuenta, además del conjunto de soluciones óptimas (frente de Pareto), el conjunto de soluciones casi-óptimas no dominadas en su vecindad. La inclusión de estas soluciones en la etapa de decisiones aumenta la diversidad del conjunto de soluciones relevantes disponibles, permitiendo elegir la solución final de una manera más informada.

La motivación principal de obtener las soluciones casi-óptimas no dominadas en su vecindad es obtener las soluciones, que siendo similares en el espacio de objetivos, difieren significativamente en el espacio de parámetros. Obtener dos soluciones similares en el espacio de parámetros (vecinas), donde una domina ligeramente a otra, no aporta nueva información relevante al diseñador. El diseñador elegirá siempre la mejor de ellas, ya que tendrán características similares, siendo una de ellas mejor en los objetivos de diseño. Sin embargo, obtener dos soluciones significativamente diferentes en el espacio de parámetros (no vecinas), donde una domina ligeramente a otra, si aporta nueva información relevante al diseñador. Ambas alternativas, teniendo prestaciones similares en los objetivos de diseño, tendrán presumiblemente un comportamiento significativamente diferente (debido a sus diferencias en los parámetros). Por ello, consideramos que nuestro conjunto de interés es el formado por las soluciones óptimas y casi-óptimas no dominadas en su vecindad. Este conjunto aumenta la diversidad de soluciones para el diseñador, sin aumentar en exceso el número de ellas.

La segunda aportación relevante de la tesis recae sobre el nuevo algoritmo nevMOGA. Existen en la literatura algoritmos destinados a la localización de alternativas casi-óptimas. Dichos algoritmos discretizan únicamente el espacio de objetivos. Con este enfoque, una solución casi-óptima puede ser eliminada sin tener en cuenta su ubicación en el espacio de parámetros, y por lo tanto, las alternativas casi-óptimas que tienen características distintas (potencialmente útiles) podrían ser descuidadas. Del mismo modo, dichos algoritmos obtendrán soluciones casi-óptimas, que bajo nuestro enfoque, no aportan información relevante al diseñador (casi-óptimas dominadas en su vecindad). Por estas razones, se desarrolló nevMOGA, capaz de caracterizar nuestro conjunto de interés.

nevMOGA se ha comparado con otros dos algoritmos (búsqueda aleatoria y búsqueda exhaustiva) sobre dos *benchmarks*, con el objetivo de evaluar su rendimiento. No se ha comparado con ningún otro algoritmo evolutivo existente, destinado a la localización de alternativas casi-óptimas, debido a que ninguno de ellos busca caracterizar nuestro conjunto de interés. Para realizar esta comparación, se ha empleado la distancia media de *Hausdorff* ( $\Delta_p$ ), una métrica que puede medir la convergencia y diversidad de cada aproximación, al conjunto ideal, en el espacio de objetivos y parámetros. Además, se ha realizado un análisis estadístico. Los resultados revelan el rendimiento superior de nevMOGA, al caracterizar el conjunto de soluciones óptimas y casi-óptimas no dominadas en su vecindad, en ambos espacios. Estos resultados se han publicado en un artículo de revista [64].

La tercera aportación está relacionada con el estudio de los problemas de optimización multiobjetivo en los que obtener el conjunto de interés puede resultar especialmente útil. Serían aquellos en los que:

1. Existen objetivos de diseño formados como la agregación de objetivos parciales.
2. Existe incertidumbre en los objetivos de diseño.
3. Existen objetivos no incluidos en la etapa de optimización (debido a una decisión del propio diseñador, por limitaciones de tiempo o económicas, etc.) que se desean estudiar, tras la etapa de optimización, en la etapa de decisión.

En muchos MOP se agregan objetivos. Un ejemplo de ello, se produce en la identificación de modelos y el diseño de controladores multivariables. En estos sistemas el número de objetivos suele ser elevado. Estos objetivos se agregan habitualmente con el fin de simplificar la etapa de optimización y la fase de decisión. En este contexto, la obtención del conjunto de interés puede ser muy relevante, ya que proporciona diversidad de alternativas con rendimiento similar, pero con un comportamiento significativamente diferente (diferente compensación entre los objetivos agregados).

También es posible que los objetivos de diseño tengan incertidumbre. Cuando esto ocurre, un simple cambio en la configuración de determinados parámetros (por ejemplo, en el entorno simulación o en el escenario definido inicialmente) puede variar el frente de Pareto. Esto podría producir que soluciones casi-óptimas pueden ser óptimas con mínimas variaciones de estos parámetros de

configuración. Por esta razón, las alternativas casi-óptimas resultan especialmente relevantes cuando existe incertidumbre en los objetivos del MOP.

Estudiar en la fase de decisión objetivos no incluidos en la fase de optimización es una práctica común. Un diseñador puede decidir excluir un objetivo de la fase de optimización por diferentes motivos (importancia secundaria, desconocimiento a priori, validación, etc). Además, el estudio de objetivos no incluidos en la fase de optimización puede deberse a limitaciones económicas (por ejemplo, objetivos medidos sobre una planta real) o temporales (por ejemplo, objetivos computacionalmente costosos). En todos estos casos, el conjunto de interés puede aportar soluciones, similares en los objetivos de diseño, con rendimiento significativamente mejor en los objetivos no incluidos en la fase de optimización.

La última aportación viene de la mano de los ejemplos de identificación de modelos y diseño de controladores multivariables presentados en la tesis. Especialmente del ejemplo de identificación y control del sistema de refrigeración de un sistema  $\mu$ -CHP basado en una pila de PEMFC. Gracias a utilizar esta nueva metodología, se han podido estudiar nuevos contextos (validación, nuevo punto de funcionamiento, análisis de robustez, nuevo periodo de muestreo, degradación del sistema, etc) una vez obtenido el conjunto de soluciones óptimas y casi-óptimas no dominadas en su vecindad. Incluir todos estos escenarios en los objetivos de diseño habría sido inabordable. Este análisis ha permitido obtener nueva información relevante para el diseñador o sistema de decisión. Por lo tanto, en esta tesis se ha demostrado la relevancia de utilizar la nueva metodología presentada en ejemplos reales de ingeniería de control. Con ella, se ha podido tomar una decisión final más informada. Gran parte de estos ejemplos han sido publicados, en una revista de primer nivel, mediante el siguiente artículo ([65]).

## 6.2 Trabajos futuros

Las líneas futuras de investigación pueden estar relacionadas con los siguientes aspectos:

- Aplicar la metodología presentada en la tesis a otros ámbitos de la ingeniería.
- Mejorar computacional del algoritmo nevMOGA, diseñado en esta tesis.

- Utilización de técnicas de clustering para determinar la vecindad entre soluciones.
- Mejoras en las herramientas de visualización para la representación simultánea del frente de Pareto y soluciones casi-óptimas no dominadas en su vecindad.
- Redactar un artículo de revista con el ejemplo de identificación y control del sistema de refrigeración de un sistema  $\mu$ -CHP basado en una pila de PEMFC.

La metodología introducida en la tesis puede ser empleada en otros problemas de ingeniería. Hay dos ejemplos sobre los que ya se ha trabajado usando técnicas clásicas de optimización multiobjetivo y se prevé que el uso de nevMOGA puede ser interesante. El primer ejemplo está relacionado con el diseño de difusores de sonido (tipo Schroeder) para mejorar las prestaciones acústicas de salas [39]. En este problema, se ha visto que difusores con estructuras significativamente diferentes pueden obtener el mismo rendimiento. Por ello, las alternativas casi-óptimas no dominadas en su vecindad pueden tener una importancia aun más relevante en este problema, con soluciones de por sí multimodales. El segundo ejemplo está relacionado con el diseño de controladores en modo deslizante para aerogeneradores. En este problema, se dispone de 18 objetivos de diseño relacionados con los diferentes modos (estados) de funcionamiento del aerogenerador. Estos objetivos podrían agregarse, reduciendo el problema a un MOP de dos objetivos. Sin embargo, esta reducción se ha desestimado, debido a que muchas soluciones óptimas pasarían a ser casi-óptimas, perdiendo soluciones interesantes. Esta problemática se podría resolver empleando la metodología presentada en la tesis. De este modo, el problema de optimización y la fase de decisión quedarían simplificadas.

El algoritmo diseñado en esta tesis (nevMOGA) podría mejorarse en distintos aspectos. Por un lado, creemos que su coste computacional podría ser reducido realizando un análisis profundo de su código. Por otro lado, las técnicas de clustering [35], cada vez más empleadas, podría utilizarse para que nevMOGA defina la vecindad del problema cuando el diseñador no dispone de la información necesaria para su definición. Esto, puede ocurrir cuando las variables de decisión carecen de sentido físico.

En la resolución de MOPs, generalmente, el diseñador se ayuda de una herramienta gráfica para comparar las prestaciones de cada una de las alternativas encontradas. En este contexto, las herramientas de visualización multidimensionales toman un papel clave en problemas de más de 2-3 dimensiones, ayu-

dando al diseñador en la decisión final. En problemas donde se obtiene un conjunto de alternativas casi-óptimas, es importante observar la degradación de las soluciones casi-óptimas respecto al frente de Pareto. Extraer esta información mediante una herramienta de visualización multidimensional existente no es una tarea trivial. Por este motivo, en futuros trabajos se pretende realizar modificaciones sobre la herramienta de visualización multidimensional Level Diagrams (LD [9] [10]), ya que es una herramienta muy extendida para la resolución de MOPs. Esta nueva mejora permitirá al diseñador observar la degradación de las soluciones casi-óptimas de una forma más intuitiva.

Por último, en trabajos futuros se puede trabajar en la publicación de los resultados obtenido en los problemas de identificación de modelos y diseño de controladores del proceso de refrigeración del sistema  $\mu$ -CHP. En estos problemas descritos en la tesis se pone de manifiesto la relevancia de la metodología presentada, obteniendo un gran cantidad de información relevante, que no hubiera sido encontrada con un enfoque multiobjetivo clásico.



## Apéndice A

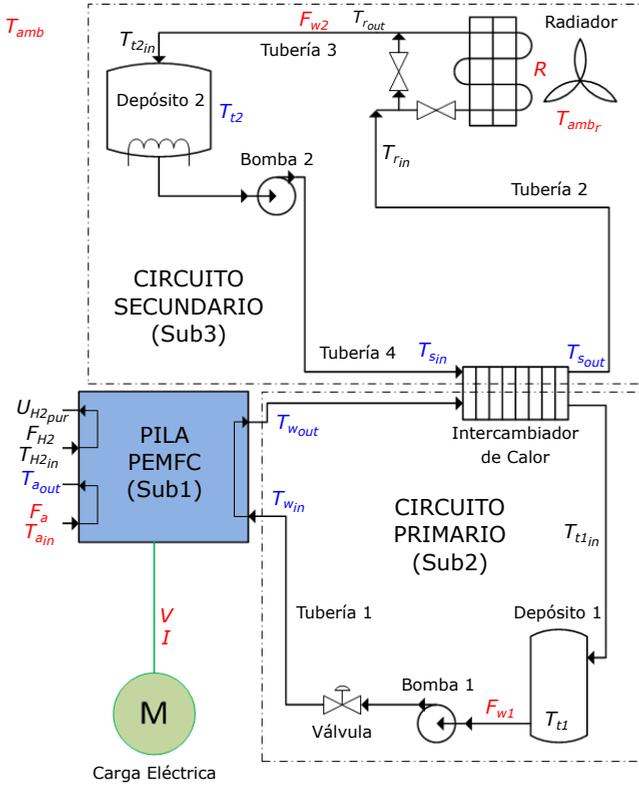
# Modelo de un sistema de refrigeración para un sistema $\mu$ -CHP



## A.1 Introducción

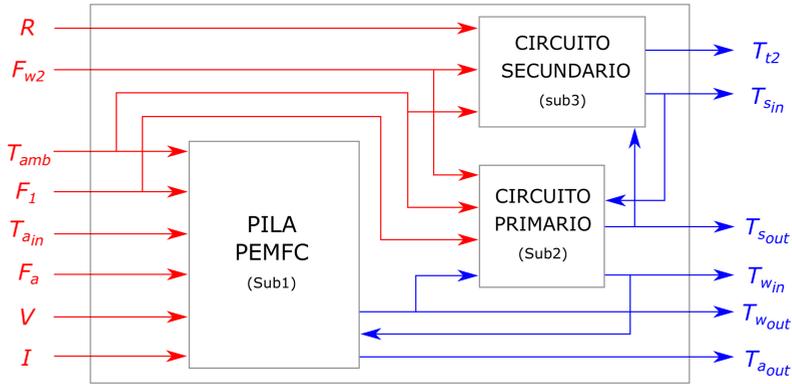
En este anexo se presenta el modelo del proceso de refrigeración de un sistema  $\mu$ -CHP (*Combined Heat and Power*) mediante una pila de tipo PEMFC (ver figura A.2). El modelo ha sido desarrollado a partir un sistema  $\mu$ -CHP real ubicado en el laboratorio de nuestro equipo de investigación, el grupo de Control Predictivo y Optimización Heurística (CPOH), de la Universitat Politècnica de València. La carga electrónica demanda energía eléctrica a la pila, requiriendo una corriente eléctrica  $i$ . La tensión de la pila  $v$  depende de la corriente demandada y de la temperatura de la pila, y viene proporcionada por el fabricante mediante su curva característica. La temperatura de la pila es medida en el agua de salida de la misma ( $T_{w_{out}}$ ). Para poder generar dicha corriente la pila ha de ser alimentada con una cantidad de hidrógeno ( $F_{H_2}$  a una temperatura de  $T_{H_2_{in}}$ ) y oxígeno ( $F_a$  a una temperatura de  $T_{a_{in}}$ ) en función de dicha demanda. Ambos caudales deben entrar a la pila con una humedad relativa del 40%. Periódicamente el circuito de hidrógeno se purga, mediante la válvula  $U_{H_2_{purgue}}$ , con el objetivo de eliminar impurezas en el interior de la pila y de evitar su encharcamiento. El aire de salida de la pila tiene una temperatura  $T_{a_{out}}$  y una humedad relativa del 100%.

La pila genera la energía eléctrica mencionada y calor. Para extraer dicho calor, el modelo consta de dos circuitos de refrigeración por agua, el circuito primario y el circuito secundario, acoplados mediante un intercambiador de calor. El agua que circula a través de la pila debe ser agua desmineralizada y con una conductividad menor de  $10\mu S/cm$ . El calor generado por la pila se extrae mediante el agua del circuito primario (con caudal  $F_{w_1}$ ) y se transfiere al circuito secundario (con caudal  $F_{w_2}$ ) mediante el intercambiador de calor, acabando finalmente en el depósito de agua caliente (depósito 2). El circuito primario está formado por una válvula y una bomba. Mediante la bomba se impulsa el agua a través de este circuito. Mediante la válvula se regula el caudal de agua que pasa a través de la pila ( $F_{w_1}$ ). Un caudal  $F_{w_1}$  grande extrae mayor calor y enfría la pila. El circuito secundario está formado por un radiador, un depósito de agua caliente y una bomba. La demanda térmica del sistema se simula mediante el radiador. Si el radiador está encendido ( $R$ ), extrae calor del circuito secundario y lo evacua al ambiente. De este modo, la temperatura del depósito de agua caliente ( $T_{t_2}$  en el depósito 2) disminuye. La bomba 2 varía el caudal del circuito secundario  $F_{w_2}$ . Si el caudal  $F_{w_2}$  aumenta, aumenta el calor transferido (desde el circuito primario al secundario mediante el intercambiador), disminuyendo la temperatura del agua de entrada a la pila  $T_{w_{in}}$ .



**Figura A.1:** Diagrama del sistema de refrigeración de un sistemas de cogeneración (generación de energía eléctrica y térmica). La salidas del modelo completo están señaladas en azul y sus entradas en rojo.

El modelo descrito corresponde al proceso de energía térmica del sistema. Por ello, la corriente eléctrica  $i$  y la tensión  $v$  que proporciona la pila son consideradas entradas, debido a que su valor influye en el proceso térmico. El caudal de hidrógeno  $F_{H2}$ , su temperatura  $T_{H2,in}$  y la válvula de purgado  $U_{H2_{purgue}}$  (solo afecta a  $F_{H2}$ ) son ignoradas debido a que su influencia resulta despreciable respecto al resto de entradas. El modelo debe ser válido para demandas de corriente ( $i$ ) entre 140 A y 200 A, y para temperaturas del depósito 2 ( $T_{t2}$ ) entre 53 y 56 °C. Este modelo completo consta de la interconexión de los tres submodelos que se presentan en este anexo (ver figura A.1).



**Figura A.2:** Diagrama de caja negra del modelo completo. El modelo está formado por la interconexión de tres submodelos (Sub1, Sub2 y Sub3). La salidas del modelo completo están señaladas en azul y sus entradas en rojo.

**Tabla A.1:** Entradas y Salidas del sistema de refrigeración de un sistema  $\mu$ -CHP.

Entradas	Descripción	Unidades
$F_a$	Caudal de aire de la pila	$m^3/s$
$T_{amb}$	Temperatura ambiente	$^{\circ}C$
$T_{a_{in}}$	Temperatura del aire de entrada a la pila	$^{\circ}C$
$v$	Tensión proporcionada por la pila	$V$
$i$	Corriente proporcionada por la pila	$A$
$F_{w_1}$	Caudal del circuito primario	$m^3/s$
$F_{w_2}$	Caudal del circuito secundario	$m^3/s$
$R$	Encendido/Apagado del radiador	<i>On/Off</i>
Salidas	Descripción	Unidades
$T_{w_{out}}$	Temperatura de salida del agua de la pila	$^{\circ}C$
$T_{w_{in}}$	Temperatura de entrada del agua de la pila	$^{\circ}C$
$T_{t2}$	Temperatura del interior del depósito 2	$^{\circ}C$
$T_{a_{out}}$	Temperatura del aire de salida de la pila	$^{\circ}C$
$T_{s_{in}}$	Temperatura del agua a la entrada del intercambiador en la carcasa	$^{\circ}C$
$T_{s_{out}}$	Temperatura del agua a la salida del intercambiador en la carcasa	$^{\circ}C$

**Tabla A.2:** Parámetros a estimar del sistema de refrigeración de un sistema  $\mu$ -CHP.

Parám.	Descripción	Unidad
$V_{t1}$	Volumen de agua que hay en el depósito 1	$m^3$
$h_{t_{min}}$	Valor mín. del parám. de transf. de calor debido a la convección forzada del tubo $h_t$	$W/K$
$h_{t_{max}}$	Valor máx. del parám. de transf. de calor debido a la convección forzada del tubo $h_t$	$W/K$
$h_{s_{min}}$	Valor mín. del parám. de transf. de calor por convección forzada de la carcasa $h_s$	$W/K$
$h_{s_{max}}$	Valor máx. del parám. de transf. de calor por convección forzada de la carcasa $h_s$	$W/K$
$V_{p1}$	Volumen de la tubería 1	$m^3$
$h_{p1_{loss}}$	Parám. de transf. de calor con el ambiente	$W/K$
$V_w$	Volumen de agua dentro de la pila	$m^3$
$V_a$	Volumen de aire dentro de la pila	$m^3$
$k_a$	Constante, producto del calor específico $c_a$ y la densidad $\rho_a$ del aire dentro de la pila.	$J/(m^3 \cdot K)$
$h_{fc2_{max}}$	Valor máx. del parám. de transf. de calor $h_{fc2}$	$W/K$
$h_{fc2_{min}}$	Valor mín. del parám. de transf. de calor $h_{fc2}$	$W/K$
$h_{fc_{loss}}$	Parámetro de transferencia de calor	$W/K$
$h_{a_{max}}$	Valor máx. del parám. de transf. de calor $h_a$	$W/K$
$h_{a_{min}}$	Valor mín. del parám. de transf. de calor $h_a$	$W/K$
$h_{w_{max}}$	Valor máx. del parám. de transf. de calor $h_w$	$W/K$
$h_{w_{min}}$	Valor mín. del parám. de transf. de calor $h_w$	$W/K$
$h_{aw}$	Parámetro de transferencia de calor	$W/K$
$calT_{w_{out}}$	Coef. de compensación de la temperatura de salida del agua de la pila	$^{\circ}C$
$calT_{a_{out}}$	Coef. de compensación de la temperatura de salida del aire de la pila	$^{\circ}C$
$V_{p4}$	Volumen de la tubería 4	$m^3$
$h_{p4_{loss}}$	Parám. de transf. de calor del agua al amb.	$W/K$
$calT_{p4_{out}}$	Coef. de compensación de la temperatura de salida de la tubería 4	$^{\circ}C$
$h_{rOFF_{min}}$	Valor mín. del parám. de transf. $h_{rOFF}$	$W/K$
$h_{rOFF_{max}}$	Valor máx. del parám. de transf. $h_{rOFF}$	$W/K$
$h_{rON_{min}}$	Valor mín. del parám. de transf. $h_{rON}$	$W/K$
$h_{rON_{max}}$	Valor máx. del parám. de transf. $h_{rON}$	$W/K$
$V_{t2}$	Volumen de agua que hay en el depósito 2	$m^3$
$V_r$	Volumen de agua que hay en radiador	$m^3$
$T_{amb_r}$	Temperatura ambiente alrededor del radiador	$^{\circ}C$

En este modelo, el símbolo  $h(W/K)$  se denomina “parámetro de transferencia de calor” y denota el producto del coeficiente de transferencia de calor general  $U$  en  $W/(K \cdot m^2)$  y el área de transferencia de calor  $A$  en  $m^2$ , es decir,  $h = UA$ .

## A.2 Submodelo 1 (Sub1)

El submodelo 1 (Sub1) está formado por un único elemento, la pila PEMFC (ver figura A.1). La pila PEMFC genera energía eléctrica y térmica. Por un lado, la energía eléctrica generada por la pila viene dada por la expresión:

$$P_e(t) = v(t)i(t) \quad (\text{A.1})$$

donde  $v$  es la tensión en bornes de la pila, e  $i$  es la corriente eléctrica generada. Por otro lado, la energía térmica generada por la pila viene dada por:

$$Q_{heat}(t) = i(t) \left[ E_c - \frac{v(t)}{n_c} \right] n_c \quad (\text{A.2})$$

donde  $E_c = 1.23V$  es la energía potencial teórica de una celda (a  $25^\circ C$  y presión atmosférica),  $n_c = 16$  es el número de celdas de la pila.

La temperatura del agua de salida de la pila evoluciona en base a la siguiente expresión:

$$\begin{aligned} k_w V_w \frac{dT_{w_{out}}(t)}{dt} &= k_w F_{w_1}(t)(T_{w_{in}}(t) - T_{w_{out}}(t)) \\ &+ Q_{heat}(t) - Q_{w_{fc}}(t) - Q_{wa}(t) \end{aligned} \quad (\text{A.3})$$

donde  $Q_{wa}$  es el caudal de aire que circula a través de la pila (ver ecuación A.4),  $Q_{w_{fc}}$  es calor que se transmite a la estructura de la pila (ver ecuación A.8),  $T_{w_{in}}(t)$  y  $T_{w_{out}}(t)$  son las temperaturas del agua de entrada y salida de la pila,  $F_{w_1}$  es el caudal de agua del circuito primario,  $V_w$  es el volumen de agua dentro de la pila y es un parámetro a ajustar, y  $k_w = c_w \rho_w$ , donde  $c_w$  es el calor específico del agua y  $\rho_w$  la densidad del agua. Se asume una temperatura del agua de  $60^\circ C$  como valor representativo en los circuitos primario y secundario. Además, tomando los valores de [52] se define  $k_w = 4112120.21 J/m^3 K$ .

$$Q_{wa}(t) = h_{fc_1}(t) \left( \frac{T_{w_{in}}(t) + T_{w_{out}}(t)}{2} - \frac{T_{a_{in}}(t) + T_{a_{out}}(t)}{2} \right) \quad (\text{A.4})$$

donde  $T_{a_{in}}(t)$  y  $T_{a_{out}}(t)$  son las temperaturas del aire de entrada y salida de la pila,  $h_{fc_1}$  es el parámetro de transferencia de calor (ver ecuación A.5).

$$h_{fc_1}(t) = \frac{1}{\frac{1}{h_w(t)} + \frac{1}{h_{aw}} + \frac{1}{h_a(t)}} \quad (\text{A.5})$$

donde  $h_{aw}$  (parámetro a ajustar),  $h_a$  (ver ecuación A.6) y  $h_w$  (ver ecuación A.7) son parámetros de transferencia de calor.

$$h_a(t) = \frac{h_{a_{max}} - h_{a_{min}}}{F_{a_{max}} - F_{a_{min}}} (6 \cdot 10^4 F_a(t) - F_{a_{min}}) + h_{a_{min}} \quad (\text{A.6})$$

donde  $F_a$  es el caudal de aire,  $F_{a_{min}}$  y  $F_{a_{max}}$  son los caudales mínimo y máximo de  $F_a$  ( $0.0020 \text{ m}^3/\text{s}$  y  $0.0027 \text{ m}^3/\text{s}$ , respectivamente), y  $h_{a_{min}}$  y  $h_{a_{max}}$  son los valores mínimo y máximo de  $h_a$  y son parámetros a ajustar.

$$h_w(t) = \frac{h_{w_{max}} - h_{w_{min}}}{F_{w_{1max}} - F_{w_{1min}}} (6 \cdot 10^4 F_{w_1}(t) - F_{w_{1min}}) + h_{w_{min}} \quad (\text{A.7})$$

donde  $h_{w_{min}}$  y  $h_{w_{max}}$  son los valores mínimo y máximo de  $h_w$  y son parámetros a ajustar, y  $F_{w_{1min}}$  y  $F_{w_{1max}}$  son los caudales mínimo y máximo de  $F_{w_1}$  ( $2.9005 \cdot 10^{-5} \text{ m}^3/\text{s}$  y  $1.1657 \cdot 10^{-4} \text{ m}^3/\text{s}$  respectivamente).

$$Q_{wfc}(t) = h_{fc_2}(t) \left( \frac{T_{w_{in}}(t) + T_{w_{out}}(t)}{2} - T_{fc}(t) \right) \quad (\text{A.8})$$

donde  $T_{fc}$  es la temperatura de la estructura de la pila y  $h_{fc_2}$  es un parámetro de transferencia de calor (ver ecuación A.9).

$$h_{fc_2}(t) = \frac{h_{fc_{2max}} - h_{fc_{2min}}}{F_{w_{1max}} - F_{w_{1min}}} (6 \cdot 10^4 F_{w_1}(t) - F_{w_{1min}}) + h_{fc_{2min}} \quad (\text{A.9})$$

donde  $h_{fc_{2min}}$  y  $h_{fc_{2max}}$  son los valores mínimo y máximo de  $h_{fc_2}$ , y son parámetros a ajustar.

La temperatura del volumen de aire en el interior de la pila aumenta el calor transferido desde el agua hasta el aire ( $Q_{wa}$ ). La temperatura del aire de salida de la pila ( $T_{a_{out}}$ ) viene dada por la siguiente expresión:

$$k_a V_a \frac{dT_{a_{out}}(t)}{dt} = (k_a F_a(t) T_{a_{in}}(t) - k_a F_a(t) T_{a_{out}}(t)) + Q_{wa}(t) \quad (\text{A.10})$$

donde  $V_a$  es el volumen de aire dentro de la pila y  $k_a$  es una constante (producto del calor específico  $c_a$  y la densidad  $\rho_a$  del aire dentro de la pila), y ambos son parámetros a ajustar.

La temperatura de la estructura metálica de la pila ( $T_{fc}$ ) aumentará por el calor transferido desde el agua hasta la propia pila ( $Q_{wfc}$ ):

$$C_{fc} \frac{dT_{fc}(t)}{dt} = Q_{wfc}(t) - h_{fc_{loss}}(T_{fc}(t) - T_{amb}(t)) \quad (\text{A.11})$$

donde  $C_{fc} = 12941.5 \text{ J/K}$  es la capacidad calorífica de la estructura de la pila (producto de la masa de la pila  $14.3 \text{ kg}$  y el calor específico del aluminio  $905 \text{ J/kg} \cdot \text{K}$ ),  $T_{amb}$  es la temperatura ambiente, y  $h_{fc_{loss}}$  es un parámetro de transferencia de calor que es un parámetro a ajustar.

Además, en este submodelo se añaden dos coeficientes de calibración  $cal_{T_{w_{out}}}$  y  $cal_{T_{a_{out}}}$  debido a errores de calibración en los sensores de temperatura de salida del agua y aire. Estos coeficientes se han añadido debido a que se han observado discrepancias entre las medidas reales y la estimadas mediante dichos sensores.

Con todo ello, el submodelo 1 (Sub1) tiene como entradas:  $F_a$ ,  $T_{amb}$ ,  $T_{a_{in}}$ ,  $v$ ,  $i$ ,  $T_{w_{in}}$  y  $F_{w1}$ ; y como salidas:  $T_{w_{out}}$  y  $T_{a_{out}}$ . Los parámetros a ajustar de este submodelo son:  $V_w$ ,  $V_a$ ,  $k_a$ ,  $h_{fc_{2max}}$ ,  $h_{fc_{2min}}$ ,  $h_{fc_{loss}}$ ,  $h_{amax}$ ,  $h_{amin}$ ,  $h_{wmax}$ ,  $h_{wmin}$ ,  $h_{aw}$ ,  $cal_{T_{w_{out}}}$  y  $cal_{T_{a_{out}}}$ .

### A.3 Submodelo 2 (Sub2)

El submodelo 2 (Sub2) está formado por tres elementos. En primer lugar, el depósito 1 y la tubería 1 del circuito primario. En segundo lugar, el intercambiador de calor que está diseñado para transferir calor entre dos fluidos.

La temperatura de salida del depósito 1  $T_{t1}(t)$  (no se considera pérdida de calor con el ambiente) viene dada por:

$$k_w V_{t1} \frac{dT_{t1}(t)}{dt} = k_w F_{w1}(t)(T_{t1_{in}}(t) - T_{t1}(t)) \quad (\text{A.12})$$

donde  $V_{t1}$  es el volumen de agua que hay en el depósito, que es constante y es un parámetro a ajustar,  $T_{t1in}$  y  $T_{t1}$  son las temperaturas de agua que entra y sale del depósito.

La temperatura del agua de la salida de la tubería 1  $T_{win}$  (es también la temperatura del agua de entrada de la pila) viene dada por la siguiente expresión:

$$k_w V_{p1} \frac{dT_{win}(t)}{dt} = k_w F_{w1}(t)(T_{t1}(t) - T_{win}(t)) - Q_{p1loss}(t) \quad (\text{A.13})$$

donde  $V_{p1}$  es el volumen de tubería 1, que se desconoce y es un parámetro a ajustar, y  $Q_{p1loss}$  es el flujo de calor con el ambiente y viene dado por:

$$Q_{p1loss}(t) = h_{p1loss} \left( \frac{T_{t1}(t) + T_{win}(t)}{2} - T_{amb}(t) \right) \quad (\text{A.14})$$

donde  $h_{p1loss}$  es el parámetro de transferencia de calor con el ambiente y es un parámetro a ajustar.

El intercambiador de calor está formado por dos zonas independientes: tubo y carcasa. A través del tubo circula el agua del circuito primario ( $F_{w1}$ ) y a través de la carcasa circula el agua del circuito secundario ( $F_{w2}$ ). Así, se transfiere el calor desde el circuito primario hasta el secundario mediante un flujo paralelo. La temperatura de salida del intercambiador ( $T_{t1in}(t)$  en el tubo, es también la temperatura de entrada al depósito 1, y  $T_{sout}(t)$  en la carcasa) viene dada por la siguiente expresión:

$$k_w V_t \frac{dT_{t1in}(t)}{dt} = k_w F_{w1}(t)(T_{wout}(t) - T_{t1in}(t)) - h_{ts}(t)\Delta\bar{T}_{ts}(t) \quad (\text{A.15})$$

y

$$k_w V_s \frac{dT_{sout}(t)}{dt} = k_w F_{w2}(t)(T_{sin}(t) - T_{sout}(t)) + h_{ts}(t)\Delta\bar{T}_{ts}(t) \quad (\text{A.16})$$

donde  $T_{wout}$  (es también la temperatura de salida del agua de la pila) y  $T_{sin}(t)$  son las temperaturas del agua a la entrada del intercambiador en el tubo y la carcasa,  $\Delta\bar{T}_{ts}$  es el gradiente de temperatura entre el tubo y la carcasa (ver ecuación A.18),  $V_t = 1.8 \cdot 10^{-4} m^3$  y  $V_s = 1.8 \cdot 10^{-4} m^3$  son los volúmenes de agua en la parte del tubo y en la carcasa (calculados teóricamente a partir de las dimensiones del intercambiador), y  $h_{ts}$  es el parámetro de transferencia de calor global del intercambiador de calor y viene dado por:

$$h_{ts}(t) = \frac{n_{ch}}{\frac{1}{h_t(t)} + \frac{1}{h_e} + \frac{1}{h_s(t)}} \quad (\text{A.17})$$

donde  $n_{ch} = 9$  es el número de canales del intercambiador de calor,  $h_t$  es el parámetro de transferencia de calor debido a la convección forzada del tubo (ver ecuación A.19),  $h_e = 216.72W/K$  es el parámetro de transferencia de calor por conducción y es un parámetro a estimar (calculado de forma teórica), y  $h_s$  es el parámetro de transferencia de calor por convección forzada de la carcasa (ver ecuación A.20).

$$\Delta\bar{T}_{ts}(t) = \frac{(T_{t1in}(t) - T_{sout}(t)) - (T_{wout}(t) - T_{sin}(t))}{\ln \frac{(T_{t1in}(t) - T_{sout}(t))}{(T_{wout}(t) - T_{sin}(t))}} \quad (\text{A.18})$$

$$h_t(t) = \frac{h_{tmax} - h_{tmin}}{F_{w1max} - F_{w1min}} (6 \cdot 10^4 F_{w1}(t) - F_{w1min}) + h_{tmin} \quad (\text{A.19})$$

donde  $h_{tmin}$  y  $h_{tmax}$  son los valores mínimo y máximo de  $h_t$ , y son parámetros a ajustar.

$$h_s(t) = \frac{h_{smax} - h_{smin}}{F_{w2max} - F_{w2min}} (6 \cdot 10^4 F_{w2}(t) - F_{w2min}) + h_{smin} \quad (\text{A.20})$$

donde  $h_{smin}$  y  $h_{smax}$  son los valores mínimo y máximo de  $h_s$ , y son parámetros a ajustar, y  $F_{w2min} = 4.2137 \cdot 10^{-5} m^3/s$  y  $F_{w2max} = 1.5493 \cdot 10^{-4} m^3/s$  son los caudales mínimo y máximo de  $F_{w2}$ .

Con todo ello, el submodelo 2 (Sub2) tiene como entradas:  $T_{wout}$ ,  $T_{sin}$ ,  $F_{w1}$ ,  $F_{w2}$  y  $T_{amb}$ ; y como salidas:  $T_{sout}$  y  $T_{win}$ . Los parámetros a ajustar de este submodelo son:  $V_{t1}$ ,  $h_{tmin}$ ,  $h_{tmax}$ ,  $h_{smin}$ ,  $h_{smax}$ ,  $V_{p1}$  y  $h_{p1loss}$ .

## A.4 Submodelo 3 (Sub3)

El submodelo 3 (Sub3) está formado por un cinco elementos: el radiador, y el depósito 2, y las tuberías 2, 3 y 4. La temperatura de salida del radiador  $T_{rout}(t)$  viene dada por la expresión:

$$k_w V_r \frac{dT_{r_{out}}(t)}{dt} = k_w F_{w_2}(t)(T_{r_{in}}(t) - T_{r_{out}}(t)) - Q_r(t) \quad (\text{A.21})$$

donde  $V_r$  es el volumen de agua que hay en el radiador, y un parámetro a ajustar,  $T_{r_{in}}(t)$  y  $T_{r_{out}}(t)$  son las temperaturas de entrada y salida del agua en el radiador, y  $Q_r$  es la energía térmica saliente del radiador y viene dada por:

$$Q_r(t) = h_r(t) \left( \frac{T_{r_{out}}(t) + T_{r_{in}}(t)}{2} - T_{amb,r} \right) \quad (\text{A.22})$$

donde  $T_{amb,r}$  es la temperatura ambiente alrededor del radiador que es un parámetro a estimar (debido a que no se ha medido), y  $h_r$  es el parámetro de transferencia de calor del radiador y viene dado por:

$$h_r(t) = \begin{cases} h_{r_{ON}}(t), & \text{para } R(t) = 1 \text{ (Radiador Encendido)} \\ h_{r_{OFF}}(t), & \text{para } R(t) = 0 \text{ (Radiador Apagado)} \end{cases} \quad (\text{A.23})$$

donde  $h_{r_{ON}}$  y  $h_{r_{OFF}}$  son los parámetros de transferencia por convección natural (ver ecuación A.24) y convección forzada (ver ecuación A.25).

$$h_{r_{ON}}(t) = \frac{h_{r_{ONmax}} - h_{r_{ONmin}}}{F_{w_{2max}} - F_{w_{2min}}} (6 \cdot 10^4 F_{w_2}(t) - F_{w_{2min}}) + h_{r_{ONmin}} \quad (\text{A.24})$$

donde  $h_{r_{ONmin}}$  y  $h_{r_{ONmax}}$  son los valores mínimo y máximo de  $h_{r_{ON}}$ , y son parámetros a ajustar.

$$h_{r_{OFF}}(t) = \frac{h_{r_{OFFmax}} - h_{r_{OFFmin}}}{F_{w_{2max}} - F_{w_{2min}}} (6 \cdot 10^4 F_{w_2}(t) - F_{w_{2min}}) + h_{r_{OFFmin}} \quad (\text{A.25})$$

donde  $h_{r_{OFFmin}}$  y  $h_{r_{OFFmax}}$  son los valores mínimo y máximo de  $h_{r_{OFF}}$ , y son parámetros a estimar.

La temperatura del agua del depósito 2  $T_{t_2}(t)$  viene dada por la siguiente expresión:

$$k_w V_{t_2} \frac{dT_{t_2}(t)}{dt} = k_w F_{w_2}(t)(T_{t_{2in}}(t) - T_{t_2}(t)) \quad (\text{A.26})$$

donde  $V_{t_2}$  es el volumen de agua que hay en el depósito 2, y es un parámetro a estimar, y  $T_{t_{2in}}$  y  $T_{t_2}$  son las temperaturas de agua a la entrada y en el interior del depósito 2.

La temperatura de salida del agua de la tubería 2  $T_{r_{int}}$  (las pérdidas de calor con el ambiente son despreciables debido a la cantidad de calor que el radiador extrae del circuito secundario) viene dada por la siguiente expresión:

$$k_w V_{p2} \frac{dT_{r_{in}}(t)}{dt} = k_w F_{w2}(t)(T_{s_{out}}(t) - T_{r_{in}}(t)) \quad (\text{A.27})$$

donde  $V_{p2} = 7.0686 \cdot 10^{-4} m^3$  es el volumen de agua de la tubería 2 (calculado teóricamente) y  $T_{s_{out}}$  (que es también la temperatura de salida del intercambiador de calor) es la temperatura del agua a la entrada de la tubería 2.

De idéntico modo, la temperatura de salida del agua de la tubería 3  $T_{t2_{in}}$  viene dada por la siguiente expresión:

$$k_w V_{p3} \frac{dT_{t2_{in}}(t)}{dt} = k_w F_{w2}(t)(T_{r_{out}}(t) - T_{t2_{in}}(t)) \quad (\text{A.28})$$

donde  $V_{p3} = 3.5343 \cdot 10^{-4} m^3$  es el volumen de agua de la tubería 3 (calculado teóricamente), y  $T_{r_{out}}$  es la temperatura del agua a la entrada de la tubería 3.

De forma análoga, la temperatura de salida del agua de la tubería 4  $T_{s_{in}}$  (es necesario modelar las pérdidas de calor con el ambiente que eran despreciables en las tuberías 2 y 3), que es también la temperatura de entrada del agua del intercambiador de calor, viene dada por la siguiente expresión:

$$k_w V_{p4} \frac{dT_{s_{in}}(t)}{dt} = k_w F_{w2}(t)(T_{t2}(t) - T_{s_{in}}(t)) - Q_{p4_{loss}}(t) \quad (\text{A.29})$$

donde  $V_{p4}$  es el volumen de la tubería y es un parámetro a ajustar, y  $Q_{p4_{loss}}$  es el flujo de pérdida de calor con el ambiente y viene dado por:

$$Q_{p4_{loss}}(t) = h_{p4_{loss}} \left( \frac{T_{t2}(t) + T_{s_{in}}(t)}{2} - T_{amb}(t) \right) \quad (\text{A.30})$$

donde  $h_{p4_{loss}}$  es el parámetro de transferencia de calor desde el agua hasta el ambiente, que se asume constante y es un parámetro a ajustar.

Además, del mismo modo que el submodelo 1, para compensar un posible error de calibración en la medición de la temperatura de salida de la tubería 4, se incluyó un coeficiente de compensación  $cal_{T_{p4_{out}}}$ , que es un parámetro a estimar.

De este modo, el submodelo 3 (Sub3) está formado por seis entradas:  $T_{s_{out}}$ ,  $F_{w_2}$ ,  $R$ ,  $T_{amb}$  y  $T_{amb_r}$ ; y una salida:  $T_{s_{in}}$ . Los parámetros a ajustar de este submodelo son:  $h_{rOFFmin}$ ,  $h_{rOFFmax}$ ,  $h_{rONmin}$ ,  $h_{rONmax}$ ,  $V_{t2}$ ,  $V_r$ ,  $T_{amb_r}$ ,  $V_{p4}$ ,  $h_{p4loss}$  y  $cal_{T_{p4out}}$ .

## A.5 Modelo completo

El modelo completo resulta de la interconexión de los tres submodelos descritos anteriormente (Sub1, Sub2 y Sub3). Este modelo del sistema de refrigeración completo se observa en las figuras A.1 y A.2. Las entradas y salidas del modelo se observan en la tabla A.1. Además, este modelo consta de 30 parámetros que se describen en la tabla A.2.

# Bibliografía

- [1] C. W. Ahn. *Advances in evolutionary algorithms*. Springer, 2006 (vid. pág. 13).
- [2] S. M. C. Ang, E. S. Fraga, N. P. Brandon, N. J. Samsatli y D. J. Brett. «Fuel cell systems optimisation—Methods and strategies». En: *International Journal of Hydrogen Energy* 36.22 (2011), págs. 14678-14703 (vid. pág. 119).
- [3] T. Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996 (vid. págs. 13, 17).
- [4] J. Bader y E. Zitzler. «HypE: An algorithm for fast hypervolume-based many-objective optimization». En: *Evolutionary computation* 19.1 (2011), págs. 45-76 (vid. págs. 20, 34).
- [5] S. Bechikh, R. Datta y A. Gupta. *Recent advances in evolutionary multi-objective optimization*. Vol. 20. Springer, 2016 (vid. pág. 13).
- [6] R. Beith. *Small and micro combined heat and power (CHP) systems: advanced design, performance, materials and applications*. Elsevier, 2011 (vid. pág. 119).

- [7] G. Bejarano, J. A. Alfaya, D. Rodríguez, F. Morilla y M. G. Ortega. «Benchmark for PID control of refrigeration systems based on vapour compression». En: *IFAC-PapersOnLine* 51.4 (2018), págs. 497-502 (vid. págs. [102-104](#)).
- [8] H.-G. Beyer y B. Sendhoff. «Robust optimization—a comprehensive survey». En: *Computer methods in applied mechanics and engineering* 196.33-34 (2007), págs. 3190-3218 (vid. pág. [27](#)).
- [9] X. Blasco, J. Herrero, J. Sanchis y M. Martínez. «A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization». En: *Information Sciences* 178.20 (2008), págs. 3908-3924 (vid. págs. [30](#), [31](#), [79](#), [155](#)).
- [10] X. Blasco, J. M. Herrero, G. Reynoso-Meza y M. A. M. Iranzo. «Interactive tool for analyzing multiobjective optimization results with level diagrams». En: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM. 2017, págs. 1689-1696 (vid. págs. [31](#), [155](#)).
- [11] D. B. Carr, R. J. Littlefield, W. Nicholson y J. Littlefield. «Scatterplot matrix techniques for large N». En: *Journal of the American Statistical Association* 82.398 (1987), págs. 424-436 (vid. pág. [31](#)).
- [12] P. C. Chang, S. H. Chen, Q. Zhang y J. L. Lin. «MOEA/D for flowshop scheduling problems». En: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE. 2008, págs. 1433-1438 (vid. pág. [19](#)).
- [13] C. A. C. Coello. «Evolutionary multi-objective optimization: basic concepts and some applications in pattern recognition». En: *Mexican Conference on Pattern Recognition*. Springer. 2011, págs. 22-33 (vid. págs. [13](#), [17](#)).
- [14] C. A. C. Coello. «Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art». En: *Computer methods in applied mechanics and engineering* 191.11-12 (2002), págs. 1245-1287 (vid. pág. [26](#)).

- 
- [15] C. A. C. Coello y N. C. Cortés. «Solving multiobjective optimization problems using an artificial immune system». En: *Genetic Programming and Evolvable Machines* 6.2 (2005), págs. 163-190 (vid. [pág. 23](#)).
- [16] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen y col. *Evolutionary algorithms for solving multi-objective problems*. Vol. 5. Springer, 2007 (vid. [págs. 3, 13](#)).
- [17] D. W. Corne, N. R. Jerram, J. D. Knowles y M. J. Oates. «PESA-II: Region-based selection in evolutionary multiobjective optimization». En: *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc. 2001, págs. 283-290 (vid. [pág. 18](#)).
- [18] D. W. Corne y J. D. Knowles. «Techniques for highly multiobjective optimisation: some nondominated points are better than others». En: *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM. 2007, págs. 773-780 (vid. [pág. 26](#)).
- [19] C. Cruz, J. R. González y D. A. Pelta. «Optimization in dynamic environments: a survey on problems, methods and measures». En: *Soft Computing* 15.7 (2011), págs. 1427-1448 (vid. [pág. 27](#)).
- [20] S. Das, S. Maity, B.-Y. Qu y P. N. Suganthan. «Real-parameter evolutionary multimodal optimization – A survey of the state-of-the-art». En: *Swarm and Evolutionary Computation* 1.2 (2011), págs. 71-88 (vid. [pág. 4](#)).
- [21] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Vol. 16. John Wiley & Sons, 2001 (vid. [pág. 3](#)).
- [22] K. Deb, S. Agrawal, A. Pratap y T. Meyarivan. «A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II». En: *International Conference on Parallel Problem Solving From Nature*. Springer. 2000, págs. 849-858 (vid. [págs. 18, 23, 33](#)).
- [23] K. Deb y A. Saha. «Finding multiple solutions for multimodal optimization problems using a multi-objective evolutionary approach». En: *Proceedings of the 12th annual conference on genetic and evolutionary computation*. ACM. 2010, págs. 447-454 (vid. [pág. 4](#)).

- [24] J. Derrac, S. García, D. Molina y F. Herrera. «A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms». En: *Swarm and Evolutionary Computation* 1.1 (2011), págs. 3-18 (vid. pág. 56).
- [25] P. E. Dodds, I. Staffell, A. D. Hawkes, F. Li, P. Grünewald, W. McDowall y P. Ekins. «Hydrogen and fuel cell technologies for heating: a review». En: *International journal of hydrogen energy* 40.5 (2015), págs. 2065-2083 (vid. pág. 119).
- [26] M. Dorigo y M. Birattari. *Ant colony optimization*. Springer, 2010 (vid. pág. 23).
- [27] R. Eberhart y J. Kennedy. «A new optimizer using particle swarm theory». En: *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. Ieee. 1995, págs. 39-43 (vid. pág. 23).
- [28] A. Elhossini, S. Areibi y R. Dony. «Strength Pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization». En: *Evolutionary Computation* 18.1 (2010), págs. 127-156 (vid. pág. 21).
- [29] H. R. Ellamla, I. Staffell, P. Bujlo, B. G. Pollet y S. Pasupathi. «Current status of fuel cell based combined heat and power systems for residential sector». En: *Journal of Power Sources* 293 (2015), págs. 312-328 (vid. pág. 119).
- [30] M. Farina, K. Deb y P. Amato. «Dynamic multiobjective optimization problems: test cases, approximations, and applications». En: *IEEE Transactions on evolutionary computation* 8.5 (2004), págs. 425-442 (vid. pág. 27).
- [31] C. M. Fonseca y P. J. Fleming. «Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation». En: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 28.1 (1998), págs. 26-37 (vid. pág. 26).
- [32] C. M. Fonseca, P. J. Fleming y col. «Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization.» En: *Icga*. Vol. 93. July. Citeseer. 1993, págs. 416-423 (vid. págs. 20, 22, 23).

- 
- [33] J. J. V. García, V. G. Garay, E. I. Gordo, F. A. Fano y M. L. Sukia. «Intelligent Multi-Objective Nonlinear Model Predictive Control (iMO-NMPC): Towards the ?on-line?optimization of highly complex control problems». En: *Expert systems with applications* 39.7 (2012), págs. 6527-6540 (vid. págs. [17](#), [85](#)).
- [34] G. C. Goodwin, S. F. Graebe, M. E. Salgado y col. *Control system design*. Upper Saddle River, NJ: Prentice Hall, 2001 (vid. pág. [3](#)).
- [35] J. Handl y J. Knowles. «An evolutionary approach to multiobjective clustering». En: *IEEE transactions on Evolutionary Computation* 11.1 (2007), págs. 56-76 (vid. pág. [154](#)).
- [36] A. Hawkes, I. Staffell, D. Brett y N. Brandon. «Fuel cells for micro-combined heat and power generation». En: *Energy & Environmental Science* 2.7 (2009), págs. 729-744 (vid. págs. [119](#), [120](#)).
- [37] J. M. Herrero Durá. «Identificación Robusta de Sistemas no Lineales mediante Algoritmos Evolutivos». Tesis doct. Editorial Universitat Politècnica de València, 2006 (vid. págs. [17](#), [43](#), [49](#), [55](#), [61](#), [90](#), [98](#), [108](#), [113](#), [125](#), [135](#)).
- [38] J. M. Herrero, S. García-Nieto, X. Blasco, V. Romero-García, J. V. Sánchez-Pérez y L. Garcia-Raffi. «Optimization of sonic crystal attenuation properties by ev-MOGA multiobjective evolutionary algorithm». En: *Structural and Multidisciplinary Optimization* 39.2 (2009), pág. 203 (vid. págs. [8](#), [21](#), [45](#)).
- [39] J. Herrero, X. Blasco, J. V. Sánchez-Pérez y J. Redondo. «Design of sound phase diffusers by means of multiobjective optimization approach using ev-MOGA evolutionary algorithm». En: *Structural and Multidisciplinary Optimization* 53.4 (2016), págs. 861-879 (vid. pág. [154](#)).
- [40] J. Herrero, G. Reynoso-Meza, C. Ramos y X. Blasco. «Considerations on loop pairing in MIMO processes. A multi-criteria analysis». En: *IFAC-PapersOnLine* 50.1 (2017), págs. 4454-4459 (vid. pág. [89](#)).
- [41] A. Inselberg y B. Dimsdale. «Parallel coordinates: a tool for visualizing multi-dimensional geometry». En: *Proceedings of the 1st conference on*

- Visualization'90*. IEEE Computer Society Press. 1990, págs. 361-378 (vid. pág. 31).
- [42] H. Ishibuchi y T. Murata. «A multi-objective genetic local search algorithm and its application to flowshop scheduling». En: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 28.3 (1998), págs. 392-403 (vid. pág. 21).
- [43] R. Jamshidi, S. F. Ghomi y B. Karimi. «Multi-objective green supply chain optimization with a new hybrid memetic algorithm using the Taguchi method». En: *Scientia Iranica* 19.6 (2012), págs. 1876-1886 (vid. pág. 21).
- [44] A. Jaszkievicz. «On the performance of multiple-objective genetic local search on the 0/1 knapsack problem—a comparative experiment». En: *IEEE Transactions on Evolutionary Computation* 6.4 (2002), págs. 402-412 (vid. pág. 21).
- [45] S. Jiang, Y.-S. Ong, J. Zhang y L. Feng. «Consistencies and contradictions of performance metrics in multiobjective optimization.» En: *IEEE Trans. Cybernetics* 44.12 (2014), págs. 2391-2404 (vid. págs. 31, 32).
- [46] J. Klemes, F. Friedler, I. Bulatov y P. Varbanov. *Sustainability in the process industry: integration and optimization (Green Manufacturing & Systems Engineering)*. McGraw-Hill Professional, New York, 2010 (vid. pág. 3).
- [47] J. D. Knowles y D. W. Corne. «M-PAES: A memetic algorithm for multiobjective optimization». En: *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*. Vol. 1. IEEE. 2000, págs. 325-332 (vid. pág. 21).
- [48] V. V. Kumar, V. Rao y M. Chidambaram. «Centralized PI controllers for interacting multivariable processes by synthesis method». En: *Isa Transactions* 51.3 (2012), págs. 400-409 (vid. pág. 101).
- [49] B.-B. Li y L. Wang. «A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling». En: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37.3 (2007), págs. 576-591 (vid. pág. 21).

- 
- [50] H. Li y Q. Zhang. «Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II». En: *IEEE Transactions on evolutionary computation* 13.2 (2009), págs. 284-302 (vid. págs. 19, 34).
- [51] X. Li, A. Engelbrecht y M. G. Epitropakis. «Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization». En: *RMIT University, Evolutionary Computation and Machine Learning Group, Australia, Tech. Rep* (2013) (vid. pág. 59).
- [52] J. H. Lienhard IV y J. H. Lienhard V. *A heat transfer textbook*. Courier Corporation, 2012 (vid. pág. 163).
- [53] P. Loridan. « $\varepsilon$ -solutions in vector minimization problems». En: *Journal of Optimization Theory and Applications* 43.2 (1984), págs. 265-276 (vid. pág. 27).
- [54] M. Lozano, D. Molina y F. Herrera. «Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems». En: *Soft Computing* 15.11 (2011), págs. 2085-2087 (vid. pág. 27).
- [55] S. Martinez, G. Michaux, P. Salagnac y J.-L. Bouvier. «Micro-combined heat and power systems (micro-CHP) based on renewable energy sources». En: *Energy Conversion and Management* 154 (2017), págs. 262-285 (vid. pág. 119).
- [56] J. R. Martins y A. B. Lambe. «Multidisciplinary design optimization: a survey of architectures». En: *AIAA journal* 51.9 (2013), págs. 2049-2075 (vid. pág. 27).
- [57] T. Mc Avoy, Y. Arkun, R. Chen, D. Robinson y P. D. Schnelle. «A new approach to defining a dynamic relative gain». En: *Control Engineering Practice* 11.8 (2003), págs. 907-914 (vid. págs. 87, 89).
- [58] Y. Mei, K. Tang y X. Yao. «Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem». En: *IEEE Transactions on Evolutionary Computation* 15.2 (2011), págs. 151-165 (vid. pág. 19).

- [59] C. Meyer, R. Wood y D. Seborg. «Experimental evaluation of analytical and Smith predictors for distillation column control». En: *AIChE Journal* 25.1 (1979), págs. 24-32 (vid. pág. 93).
- [60] K. Miettinen. *Nonlinear Multiobjective Optimization*. Vol. 1. Kluwer Academic Publishers, 1998 (vid. pág. 3).
- [61] M. Munro y B. Aouni. «Group decision makers' preferences modelling within the goal programming model: An overview and a typology». En: *Journal of Multi-Criteria Decision Analysis* 19.3-4 (2012), págs. 169-184 (vid. pág. 26).
- [62] T. Murata y H. Ishibuchi. «MOGA: Multi-objective genetic algorithms». En: *IEEE international conference on evolutionary computation*. Vol. 1. 1995, págs. 289-294 (vid. pág. 17).
- [63] D. A. Notter, K. Kouravelou, T. Karachalios, M. K. Daletou y N. T. Haberland. «Life cycle assessment of PEM FC applications: electric mobility and  $\mu$ -CHP». En: *Energy & Environmental Science* 8.7 (2015), págs. 1969-1985 (vid. pág. 119).
- [64] A. Pajares, X. Blasco, J. M. Herrero y G. Reynoso-Meza. «A multi-objective genetic algorithm for the localization of optimal and nearly optimal solutions which are potentially useful: nevMOGA». En: *Complexity* 2018 (2018) (vid. pág. 152).
- [65] A. Pajares, X. Blasco, J. M. Herrero y G. Reynoso-Meza. «A new point of view in multivariable controller tuning under multiobjective optimization by considering nearly optimal solutions». En: *IEEE Access* (2019) (vid. pág. 153).
- [66] A. Pajares, X. Blasco, J. M. Herrero y R. Simarro. «Non-linear robust identification of a lead acid battery model using multiobjective evolutionary algorithms». En: *IFAC-PapersOnLine* 50.1 (2017), págs. 4466-4471 (vid. pág. 3).
- [67] A. Pajares, X. Blasco, J. M. Herrero y R. Simarro. «Using a Multiobjective Approach to Compare Multiple Design Alternatives—An Application to Battery Dynamic Model Tuning». En: *Energies* 10.7 (2017), pág. 999 (vid. pág. 3).

- 
- [68] S. Pal, B. Qu, S. Das y P. Suganthan. «Optimal synthesis of linear antenna arrays with multi-objective differential evolution». En: *Progress In Electromagnetics Research B* 21 (2010), págs. 87-111 (vid. pág. 19).
- [69] P. Palmers, T. McConnaghy, M. Steyaert y G. Gielen. «Massively multi-topology sizing of analog integrated circuits». En: *Proceedings of the Conference on Design, Automation and Test in Europe*. European Design y Automation Association. 2009, págs. 706-711 (vid. pág. 19).
- [70] V. Pareto y col. «Manual of political economy». En: (1971) (vid. pág. 14).
- [71] R. N. Pawar y S. P. Jadhav. «Design of NDT and PSO based decentralised PID controller for wood-berry distillation column». En: *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*. IEEE. 2017, págs. 719-723 (vid. pág. 93).
- [72] G. T. Pulido y C. A. C. Coello. «Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer». En: *Genetic and Evolutionary Computation Conference*. Springer. 2004, págs. 225-237 (vid. pág. 19).
- [73] R. C. Purshouse y P. J. Fleming. «On the evolutionary optimization of many conflicting objectives». En: *IEEE Transactions on Evolutionary Computation* 11.6 (2007), págs. 770-784 (vid. pág. 26).
- [74] B. Qian, L. Wang, D.-X. Huang y X. Wang. «Multi-objective no-wait flow-shop scheduling with a memetic algorithm based on differential evolution». En: *Soft Computing* 13.8-9 (2009), pág. 847 (vid. pág. 21).
- [75] B.-Y. Qu y P. N. Suganthan. «Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection». En: *Information sciences* 180.17 (2010), págs. 3170-3181 (vid. pág. 23).
- [76] S. S. Rao. *Engineering optimization: theory and practice*. John Wiley & Sons, 2009 (vid. pág. 3).
- [77] G. Reynoso Meza. «Controller tuning by means of evolutionary multiobjective optimization: a holistic multiobjective optimization design

- procedure». Tesis doct. Editorial Universitat Politècnica de València, 2014 (vid. pág. 26).
- [78] G. Reynoso-Meza, X. Blasco, J. Sanchis y J. M. Herrero. *Controller Tuning with Evolutionary Multiobjective Optimization: A Holistic Multiobjective Optimization Design Procedure*. Vol. 85. Springer, 2016 (vid. pág. 24).
- [79] G. Reynoso-Meza, J. Sanchis, X. Blasco y S. García-Nieto. «Physical programming for preference driven evolutionary multi-objective optimization». En: *Applied Soft Computing* 24 (2014), págs. 341-362 (vid. pág. 23).
- [80] G. Reynoso-Meza, J. Sanchis, X. Blasco y J. M. Herrero. «Multiobjective evolutionary algorithms for multivariable PI controller design». En: *Expert Systems with Applications* 39.9 (2012), págs. 7895-7907 (vid. pág. 3).
- [81] G. Reynoso-Meza, J. Sanchis, X. Blasco y M. Martínez. «Evolutionary algorithms for PID controller tuning: Current trends and perspectives». En: *Revista Iberoamericana de Automática e Informática Industrial* 10.3 (2013), págs. 251-268 (vid. pág. 3).
- [82] N. Riquelme, C. Von Lüken y B. Baran. «Performance metrics in multi-objective optimization». En: *Computing Conference (CLEI), 2015 Latin American*. IEEE. 2015, págs. 1-11 (vid. págs. 31-33).
- [83] S. Sanaye y H. Hajabdollahi. «Thermal-economic multi-objective optimization of plate fin heat exchanger using genetic algorithm». En: *Applied Energy* 87.6 (2010), págs. 1893-1902 (vid. pág. 17).
- [84] L. Sánchez y J. R. Villar. «Obtaining transparent models of chaotic systems with multi-objective simulated annealing algorithms». En: *Information Sciences* 178.4 (2008), págs. 952-970 (vid. pág. 23).
- [85] J. Sanchis, M. A. Martínez y X. Blasco. «Integrated multiobjective optimization and a priori preferences using genetic algorithms». En: *Information Sciences* 178.4 (2008), págs. 931-951 (vid. págs. 16, 20, 69).

- 
- [86] B. Sareni y L. Krahenbuhl. «Fitness sharing and niching methods revisited». En: *IEEE Transactions on Evolutionary computation* 2.3 (1998), págs. 97-106 (vid. pág. 47).
- [87] W. Schmittinger y A. Vahidi. «A review of the main parameters influencing long-term performance and durability of PEM fuel cells». En: *Journal of power sources* 180.1 (2008), págs. 1-14 (vid. págs. 120, 138).
- [88] O. Schütze, C. A. C. Coello y E.-G. Talbi. «Approximating the  $\varepsilon$ -efficient set of an MOP with stochastic search algorithms». En: *Mexican International Conference on Artificial Intelligence*. Springer. 2007, págs. 128-138 (vid. págs. 27, 28).
- [89] O. Schütze, C. A. C. Coello, E. Tantar y E.-G. Talbi. «Computing a finite size representation of the set of approximate solutions of an MOP». En: *arXiv preprint arXiv:0804.0581* (2008) (vid. págs. 7, 27, 28, 35).
- [90] O. Schütze, X. Esquivel, A. Lara y C. A. C. Coello. «Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization». En: *IEEE Transactions on Evolutionary Computation* 16.4 (2012), págs. 504-522 (vid. págs. 32, 35, 36).
- [91] O. Schütze, M. Vasile y C. A. C. Coello. «Approximate solutions in space mission design». En: *International Conference on Parallel Problem Solving from Nature*. Springer. 2008, págs. 805-814 (vid. págs. 28, 52).
- [92] O. Schütze, M. Vasile y C. A. C. Coello. «Computing the set of epsilon-efficient solutions in multi-objective space mission design». En: *Computing* 8 (2011) (vid. págs. 27, 30, 35, 52, 53, 57).
- [93] S. Skogestad. «Control structure design for complete chemical plants». En: *Computers & Chemical Engineering* 28.1-2 (2004), págs. 219-234 (vid. pág. 3).
- [94] S. Skogestad y C. Grimholt. «The SIMC method for smooth PID controller tuning». En: *PID Control in the Third Millennium*. Springer, 2012, págs. 147-175 (vid. pág. 97).

- [95] S. Skogestad e I. Postlethwaite. *Multivariable feedback control: analysis and design*. Vol. 2. Wiley New York, 2007 (vid. pág. 101).
- [96] K. I. Smith, R. M. Everson, J. E. Fieldsend, C. Murphy y R. Misra. «Dominance-based multiobjective simulated annealing». En: *IEEE Transactions on Evolutionary Computation* 12.3 (2008), págs. 323-342 (vid. pág. 23).
- [97] N. Srinivas y K. Deb. «Multiobjective optimization using nondominated sorting in genetic algorithms». En: *Evolutionary computation* 2.3 (1994), págs. 221-248 (vid. pág. 22).
- [98] I. Staffell. «Fuel cells for domestic heat and power: are they worth it?» Tesis doct. University of Birmingham, 2010 (vid. pág. 138).
- [99] R. Storn y K. Price. «Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces». En: *Journal of global optimization* 11.4 (1997), págs. 341-359 (vid. pág. 23).
- [100] L. E. Toothaker. *Multiple comparison procedures*. 89. Sage, 1993 (vid. pág. 56).
- [101] T. Tušar y B. Filipič. «Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the projection method». En: *IEEE Transactions on Evolutionary Computation* 19.2 (2015), págs. 225-245 (vid. pág. 30).
- [102] D. A. Van Veldhuizen. «Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations». Tesis doct. PhD thesis, Department of Electrical y Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1999 (vid. pág. 32).
- [103] D. A. Van Veldhuizen y G. B. Lamont. «Multiobjective evolutionary algorithm test suites». En: *Proceedings of the 1999 ACM symposium on Applied computing*. ACM. 1999, págs. 351-357 (vid. pág. 32).

- 
- [104] M. Vasile y M. Locatelli. «A hybrid multiagent approach for global trajectory optimization». En: *Journal of Global Optimization* 44.4 (2009), págs. 461-479 (vid. pág. 27).
- [105] T. N. L. Vu y M. Lee. «Multi-loop PI controller design based on the direct synthesis for interacting multi-time delay processes». En: *ISA transactions* 49.1 (2010), págs. 79-86 (vid. pág. 101).
- [106] L. While y L. Bradstreet. «Applying the WFG algorithm to calculate incremental hypervolumes». En: *2012 IEEE Congress on Evolutionary Computation*. IEEE. 2012, págs. 1-8 (vid. pág. 34).
- [107] D. J. White. «Epsilon efficiency». En: *Journal of Optimization Theory and Applications* 49.2 (1986), págs. 319-337 (vid. pág. 27).
- [108] R. Wood y M. Berry. «Terminal composition control of a binary distillation column». En: *Chemical Engineering Science* 28.9 (1973), págs. 1707-1717 (vid. págs. 93, 94).
- [109] D. Yang, L. Jiao y M. Gong. «Adaptive multi-objective optimization based on nondominated solutions». En: *Computational Intelligence* 25.2 (2009), págs. 84-108 (vid. pág. 21).
- [110] W.-C. Yeh y M.-C. Chuang. «Using multi-objective genetic algorithm for partner selection in green supply chain problems». En: *Expert Systems with applications* 38.4 (2011), págs. 4244-4253 (vid. pág. 17).
- [111] W. Yu, B. Li, H. Jia, M. Zhang y D. Wang. «Application of multi-objective genetic algorithm to optimize energy efficiency and thermal comfort in building design». En: *Energy and Buildings* 88 (2015), págs. 135-143 (vid. pág. 17).
- [112] X. Yu y M. Gen. *Introduction to evolutionary algorithms*. Springer Science & Business Media, 2010 (vid. pág. 13).
- [113] Y. Yuan y H. Xu. «Multiobjective flexible job shop scheduling using memetic algorithms». En: *IEEE Transactions on Automation Science and Engineering* 12.1 (2015), págs. 336-353 (vid. pág. 21).

- [114] G. Zhang y S. G. Kandlikar. «A critical review of cooling techniques in proton exchange membrane fuel cell stacks». En: *international journal of hydrogen energy* 37.3 (2012), págs. 2412-2429 (vid. pág. 120).
- [115] Q. Zhang y H. Li. «MOEA/D: A multiobjective evolutionary algorithm based on decomposition». En: *IEEE Transactions on evolutionary computation* 11.6 (2007), págs. 712-731 (vid. págs. 19, 21).
- [116] Q. Zhang, W. Liu y H. Li. «The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances». En: *2009 IEEE congress on evolutionary computation*. IEEE. 2009, págs. 203-208 (vid. pág. 19).
- [117] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan y Q. Zhang. «Multiobjective evolutionary algorithms: A survey of the state of the art». En: *Swarm and Evolutionary Computation* 1.1 (2011), págs. 32-49 (vid. págs. 19, 22).
- [118] E. Zitzler y S. Künzli. «Indicator-based selection in multiobjective search». En: *International Conference on Parallel Problem Solving from Nature*. Springer. 2004, págs. 832-842 (vid. pág. 20).
- [119] E. Zitzler, M. Laumanns y L. Thiele. «SPEA2: Improving the strength Pareto evolutionary algorithm». En: *TIK-report* 103 (2001) (vid. págs. 18, 27).
- [120] E. Zitzler y L. Thiele. «Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach». En: *IEEE transactions on Evolutionary Computation* 3.4 (1999), 257-271 (vid. págs. 22, 27, 34).
- [121] E. Zitzler y L. Thiele. «Multiobjective optimization using evolutionary algorithms—a comparative case study». En: *International conference on parallel problem solving from nature*. Springer. 1998, págs. 292-301 (vid. pág. 34).
- [122] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca y V. G. Da Fonseca. «Performance assessment of multiobjective optimizers: An analysis and review». En: *IEEE Transactions on evolutionary computation* 7.2 (2003), págs. 117-132 (vid. págs. 32, 34).