



**UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA**

**Departamento de Estadística e Investigación Operativa
Aplicadas y Calidad**

TESIS DOCTORAL

**Optimización del diseño de parámetros:
Métodos Forest-Genetic**

**Presentada por:
Adriana Villa Murillo**

**Dirigida por:
Dr. D^a. Susana San Matías Izquierdo
Dr. D. Andrés Carrión García**

Valencia, 2012

A mi angelito en el cielo.

*A Susana: ejemplo de profesionalismo
y amor familiar.*

Agradecimiento

A Dios y la Divina Pastora por sobre todas las cosas.

A mis directores de tesis, Susana San Matías Izquierdo y Andrés Carrión García, por el apoyo brindado en todo momento, sus oportunos consejos, observaciones e ideas en el desarrollo de mi formación doctoral.

A los miembros del Dpto. de Estadística e Investigación Operativa Aplicadas y Calidad de la U.P.V. quienes de una u otra forma contribuyeron con la culminación del presente trabajo; en especial a Alberto Ferrer, Sebastián Balasch, Ana Debón y Ana Aparicio por su trato cordial en los cortos pero muy valiosos momentos compartidos.

A mis compañeros de despacho Jose María, Andrés y Nora: juntos formamos un grupo ameno con ideas coherentes, y a veces no tan coherentes, que ayudaron a conducir largas horas de trabajo.

A Vicent Giner, por brindarme su sincera amistad y asesoría oportuna en el inmenso mundo .tex: “lo prometido es deuda”.

A la Universidad Centroccidental "Lisandro Alvarado" (UCLA-Venezuela) por fungir como mi ente financiador durante mi estadía en España.

A mi familia, en especial a Faride quien a pesar de la distancia estuvo siempre presente para inyectar la fé que muchas veces perdí y a mi esposo, Antonio, por "hacerme pisar tierra" en todo momento.

Finalmente a Pepe Botella, por proporcionar con su ánimo y entusiasmo el mejor anti-estrés en cada una de sus clases.

A todos, mil gracias..!!

Resumen

En el entorno altamente competitivo en que hoy se mueven las empresas, la calidad se ha convertido en la herramienta clave de supervivencia de las mismas. Hoy en día es generalmente aceptado que esa calidad debe ser conseguida desde el diseño, tanto de los productos como de los procesos. Taguchi propone la metodología del diseño robusto de parámetros con el fin de diseñar productos menos sensibles a los factores aleatorios o de ruido que hacen que varíen los parámetros que definen su calidad. Tal metodología ha sido comúnmente empleada en la industria a pesar de las grandes controversias que despertó desde sus inicios.

El presente trabajo aporta una alternativa completa y estadísticamente robusta para la mejora del diseño de parámetros considerando en primer lugar experimentos con una sola característica de calidad como respuesta y posteriormente generalizado a casos con múltiples características de calidad. Tal propuesta, definida como *Métodos Forest-Genetic*, combina herramientas de minería de datos y metaheurísticas en 3 fases: normalización, modelización y optimización.

Tanto para el caso univariante como para el caso multivariante se comparan

los resultados de forma numérica con los aportes más recientemente encontrados en la literatura mediante 4 casos de estudio diferentes. Pudimos verificar que nuestra propuesta metodológica se concentra en las variables de mayor importancia producto del proceso de modelización, lo que nos permite alcanzar importantes mejoras en cuanto al objetivo de calidad considerado en cada caso.

Abstract

In the highly competitive environment in which companies work, quality has become a key tool of survival. It is now accepted that quality must be achieved from the design of both products and processes. Taguchi proposed a methodology for robust design of parameters, to design products less sensitive to random factors or noise that cause variability in the parameters that define its quality. This methodology has been commonly used in industry despite the great controversies aroused from its inception.

This paper provides a complete and statistically robust alternative for improving parameter's design, considering firstly experiments with a single quality characteristic as response variable and subsequently it has been generalized to cases with multiple quality characteristics. This proposal, termed as Forest-Genetic Method, combines data mining tools and metaheuristics in 3 phases: normalization, modeling and optimization.

For both univariate and multivariate cases, results are numerically compared with the most recent contributions in the literature using 4 different case studies. We verified that our proposed methodology is focused on the most important variables of the product modeling process, allowing

us to achieve significant improvements in the quality objective considered in each case.

Resum

En l'entorn altament competitiu en què avui es mouen les empreses, la qualitat s'ha convertit en l'eina clau de supervivència de les mateixes i aquesta qualitat ha de ser aconseguida des del disseny, tant dels productes com dels processos. Taguchi proposa la metodologia del disseny robust de paràmetres per tal de dissenyar productes menys sensibles als factors aleatoris o de soroll que fan que variïn els paràmetres que defineixen la seva qualitat. Tal metodologia ha estat comunament emprada en la indústria malgrat les grans controvèrsies que va despertar des dels seus inicis.

Aquest treball aporta una alternativa completa i estadísticament robusta per a la millora del disseny de paràmetres considerant en primer lloc experiments amb una sola característica de qualitat com a resposta i posteriorment generalitzat a casos amb múltiples característiques de qualitat. Aquesta proposta, definida com *emph* Mètodes Forest-Genetic, combina eines de mineria de dades i metaheurístiques en 3 fases: normalització, modelització i optimització.

Tant per al cas univariant com per al cas multivariant es comparen els resultats de forma numèrica amb les aportacions més recentment trobats

a la literatura mitjançant 4 casos d'estudi diferents. Vam poder verificar que la nostra proposta metodològica es concentra en les variables de major importància producte del procés de modelització, el que els permet assolir importants millores pel que fa a l'objectiu de qualitat considerat en cada cas.

Índice general

Índice de tablas	XV
Índice de figuras	XVII
Abreviaturas	XIX
I Motivación y fundamentos teóricos	1
1. Introducción	3
1.1. El diseño robusto desde la óptica Taguchi	6
1.2. Objetivos	11
1.2.1. Objetivo general	11
1.2.2. Objetivos específicos	11
1.3. Fases para el desarrollo de la metodología propuesta	12
1.4. Organización del trabajo	14
2. El estado del arte	19
2.1. Metodología Taguchi	20
2.1.1. Relaciones S/N	26

2.1.2.	Metodología Taguchi en sistemas dinámicos	29
2.1.3.	Metodología Taguchi en sistemas dinámicos con múltiples respuestas	34
2.1.4.	Metodología Taguchi en sistemas no dinámicos	37
2.2.	Evolución de la metodología Taguchi	40
II	Estrategias para la modelización de sistemas	45
3.	Introducción	47
3.1.	Aprendizaje supervisado	48
3.2.	Árboles de clasificación y regresión	49
3.2.1.	Random Forest	58
3.2.2.	Árboles de regresión multivariante	63
3.3.	Redes Neuronales Artificiales	65
4.	Modelización para casos univariantes	73
4.1.	Esquema general de modelización	73
4.2.	Estrategias basadas en CART y RF	75
4.2.1.	Normalización de los datos	77
4.2.2.	Desarrollo y selección de los algoritmos para la modelización	78
4.3.	Estrategias basadas en ANN	85
4.4.	Casos de ilustración	87
4.4.1.	Parámetros de control	88
4.4.2.	Caso 4.1	90
4.4.3.	Caso 4.2	106
5.	Modelización para casos multivariantes	111
5.1.	Esquema general de modelización	112
5.2.	Estrategia basada en Random Forest multivariante	114

5.2.1.	Normalización y métricas de modelaje	114
5.2.2.	Desarrollo y selección de los algoritmos para la modelización	116
5.3.	Estrategias basadas en ANN	118
5.4.	Casos de ilustración	120
5.4.1.	Parámetros de control	121
5.4.2.	Caso 5.1	121
5.4.3.	Caso 5.2	129
6.	Estrategias definitivas de modelización	133
6.1.	Metodología PRED-RF	134
6.2.	Metodología MRF	138
III	Estrategias para la optimización de sistemas	141
7.	Introducción	143
7.1.	Algoritmos Genéticos	144
7.1.1.	Componentes de los Algoritmos Genéticos	146
7.2.	Funciones de deseabilidad	154
8.	Optimización para casos univariantes	161
8.1.	Componentes genéticos	162
8.1.1.	Población inicial y función Fitness	162
8.1.2.	Cruce ponderado	163
8.2.	Estrategia de optimización	164
8.3.	Casos de ilustración	166
8.3.1.	Optimización del caso 4.1	166
8.3.2.	Optimización del caso 4.2	170

9. Optimización para casos multivariantes	177
9.1. Componentes genéticos	178
9.1.1. Población inicial y función Fitness	179
9.1.2. Cruce ponderado	181
9.1.3. Funciones de mutación	181
9.1.4. Predicción de nuevos cromosomas	183
9.2. Estrategia de optimización	183
9.3. Casos de ilustración	185
9.3.1. Optimización del caso 5.1	186
9.3.2. Optimización del caso 5.2	195
10. Estrategias definitivas de optimización	203
10.1. Casos univariantes	204
10.2. Casos multivariantes	207
IV Conclusiones y futuras líneas de investigación	211
11. Conclusiones	213
11.1. Cumplimiento de los objetivos	213
11.2. Los métodos Forest-Genetic	215
12. Investigaciones futuras	221
Bibliografía	223
A. Datos casos 4.1 y 4.2	231
B. Datos casos 5.1 y 5.2	235

Índice de tablas

2.1. Matriz de diseño compuesta según la metodología Taguchi	25
2.2. Esquema de análisis según la metodología Taguchi	27
4.1. Funciones de normalización	78
4.2. Progreso de los valores de RMSE de PRED-RF frente a PRED-ANN851. Caso 4.1	100
4.3. Progreso de los valores de RMSE de PRED-RF frente a PRED-ANN, en conjuntos de entrenamiento y prueba. Caso 4.1	105
4.4. Progreso de los valores de RMSE de PRED-RF frente a ANN, en conjuntos de entrenamiento y prueba. Caso 4.2	109
5.1. Métricas de modelaje para los algoritmos MRF	115
5.2. Localización de los factores de control en L_{18}	123
5.3. Especificación de las respuestas. Caso 5.1	123
5.4. Progreso en los valores de RMSE de los algoritmos MRF_{P_p} frente a MRT. Caso 5.1	125
5.5. Progreso de MRF_{P_2} frente a otras metodologías. Caso 5.1	128
5.6. Especificación de las respuestas. Caso de ilustración 5.2	129

5.7. Comparación de los RMSE en consideración al número de iteraciones.Caso 5.2	132
7.1. Escala de deseabilidad para d	155
8.1. Medidas de importancia y pesos de los genes según PRED-RF. Caso 4.1	167
8.2. Comparación de los resultados en diferentes propuestas de optimización. Caso 4.1.	170
8.3. Medidas de importancia y pesos de los genes según PRED-RF. Caso 4.2	171
8.4. Comparación de los resultados en diferentes propuestas de optimización. Caso 4.2	175
9.1. Medidas de rendimiento según el tipo de respuesta	180
9.2. Funciones de mutación para el caso multivariante	182
9.3. Medidas de importancia y pesos de los genes según MRF_{P2} . Caso 5.1	188
9.4. ANOVA para las funciones de mutación. Caso 5.1	189
9.5. Comparación de los resultados en diferentes propuestas de optimización multivariante. Caso 5.1	195
9.6. Medidas de importancia y pesos de los genes según MRF_{P2} . Caso 5.2	197
9.7. Comparación de los resultados en diferentes propuestas de optimización multivariante. Caso 5.2	201
12.1. Conjunto de datos. Caso 4.1	232
12.2. Conjunto de datos. Caso 4.2	233
12.3. Conjunto de datos. Caso 5.1	235
12.4. Conjunto de datos. Caso 5.2	240

Índice de figuras

1.1. Fases para el desarrollo de la metodología propuesta	14
2.1. Función de pérdida de Taguchi	21
2.2. Sistemas dinámicos	30
2.3. Sistemas dinámicos con múltiples respuestas	35
3.1. Árbol de regresión para $\mathbb{X} = \{X_1, X_2\}$	51
3.2. Esquema general para validación cruzada con $k=5$	55
3.3. Algoritmo general Bootstrap	57
3.4. Algoritmo general Bagging	58
3.5. Algoritmo Random Forest para $b = 1000$	62
3.6. Funcionamiento de una neurona en ANN	66
3.7. Estructura estándar de una red neuronal	67
4.1. Esquema general de las estrategias basadas en CART y ANN	74
4.2. Estrategias basadas en CART	81
4.3. Estrategia basada en ANN	86
4.4. Funciones de normalización	91
4.5. Media y Desviación estándar por función de normalización .	92
4.6. Ajuste del parámetro de complejidad (cp). Caso 4.1	94

4.7. Ajuste del parámetro m_{try} mediante la tasa de error OOB.	
Caso 4.1	95
4.8. Comparación del RMSE de las estrategias basadas en CART. Caso 4.1	96
4.9. Árbol de regresión ajustado.	97
4.10. Comparación del RMSE de las estrategias basadas en ANN.	
Caso 4.1	98
4.11. Comparación de los rendimientos de RMSE para todas las estrategias. Caso 4.1	100
4.12. Comparación RMSE de PRED-RF en conjunto de entrenamiento y prueba. Caso 4.1	102
4.13. Comparación RMSE de PRED-ANN851 en conjunto de entrenamiento y prueba. Caso 4.1	103
4.14. Comparación RMSE de PRED-RF y PRED-ANN851. Caso 4.1	103
(a). Conjunto de entrenamiento	103
(b). Conjunto de prueba	103
4.15. Ajuste del parámetro m_{try} mediante la tasa de error OOB.	
Caso 4.2	107
4.16. Comparación RMSE de PRED-RF con ANN en conjunto de entrenamiento y prueba. Caso 4.2	109
5.1. Esquema general de las estrategias basadas en MRF y ANN	113
5.2. Estrategias basadas en MRF	117
5.3. Comparación del RMSE en las métricas de modelaje	124
5.4. Ajuste del parámetro m_{try} mediante la tasa de error OOB.	
Caso 5.1	126
5.5. Comparación de los $RMSE_{MRF_{P_j}}$ con otras propuestas	128

5.6. Ajuste del parámetro m_{try} mediante la tasa de error OOB. Caso 5.2	130
5.7. Comparación del RMSE de MRF_{P_2} con los trabajos realizados por Chang y Chen (2011)	131
6.1. Metodología PRED-RF	137
6.2. Metodología MRF	140
7.1. Esquema general de un algoritmo genético simple	147
7.2. Cruce de 1 punto	152
7.3. Cruce de 2 puntos	153
7.4. Escala de Deseabilidad	157
8.1. Comportamiento del GA para la longitud del canal de gas. Caso 4.1	169
8.2. Comportamiento del GA para el grosor del patrón epoxy. Caso 4.2	174
9.1. Prueba de Tukey para tasa de mutación. Caso 5.1	190
9.2. Prueba de Tukey para el tipo de mutación. Caso 5.1	190
9.3. Gráfico de interacciones de efectos significantes. Caso 5.1	191
9.4. Comportamiento del GA multivariante en 15 generaciones. Caso 5.1	194
9.5. Comportamiento del GA multivariante en 15 generaciones. Caso 5.2	200
10.1. Propuesta de optimización para casos univariantes	207
10.2. Propuesta de optimización para casos multivariantes	210
11.1. Método Forest-Genetic univariante	217
11.2. Método Forest-Genetic multivariante	218

Abreviaturas

ANN:	Redes neuronales artificiales
ANOVA:	Análisis de la varianza
CART:	Árboles de clasificación y regresión
DLTB:	Relación dinámica <i>mayor es mejor</i>
DNTB:	Relación dinámica <i>nominal es mejor</i>
DSTB:	Relación dinámica <i>menor es mejor</i>
GA:	Algoritmos genéticos
RF:	Random forest
LTB:	Relación estática <i>mayor es mejor</i>
NTB:	Relación estática <i>nominal es mejor</i>
MRF:	Random forest multivariante
PRED-T+VC:	Algoritmo que combina CART con validación cruzada
PRED-T+Boot:	Algoritmo que combina CART con validación Bootstrap
PRED-T+Bagg:	Algoritmo que combina CART con validación Bagging
PRED-T+RF:	Algoritmo que combina CART con RF
PRED-ANN:	Algoritmo basado en ANN
RMSE:	Raíz del error cuadrático medio
SA:	Simulated annealing
SPC:	Control estadístico de procesos
S/N:	Relaciones señal-ruido
STB:	Relación estática <i>menor es mejor</i>

Parte I

Motivación y fundamentos
teóricos

Capítulo 1

Introducción

Los avances tecnológicos y la globalización de los mercados arrastran mayores exigencias en los procesos de producción demandando mayores índices de calidad, por tanto se requiere que las tasas de fallo de los componentes que integran el proceso sean minimizados casi en su totalidad. Nace entonces la necesidad de crear productos robustos, es decir, productos diseñados de tal forma que mantienen sus características de calidad con un mínimo nivel de variabilidad. Todo ello trae como consecuencia que el diseño y fabricación de los productos cada vez sea más compleja, haciéndose necesaria la presencia de técnicos con alto nivel de formación y líderes, con conocimientos profundos del producto, que coordinen la distintas fases del proceso de producción. Así pues, en el proceso de fabricación de un producto se pueden distinguir 3 etapas:

Diseño del producto: Donde se concibe y se materializa el producto con las exigencias de calidad que tendrá.

Diseño del proceso: Se ocupa de determinar la manera en que se llevara a cabo la fabricación del producto.

Etapa de producción: Es la fase de la fabricación de los productos diseñados en la primera etapa con los métodos propuestos en la segunda etapa.

Por otro lado, existen diferentes causas que pueden afectar el funcionamiento del producto; las causas externas pueden aparecer luego que el producto esté en manos del cliente por variaciones en las condiciones de funcionamiento, en el entorno, en la forma de uso o en otros factores. Las causas internas son intrínsecas al producto debido a envejecimiento, deterioro, etc. Pero las más importantes, de entre las causas intrínsecas, son las causas relacionadas con la producción que pueden aparecer en cualquier etapa del proceso de producción, afectando a los productos salientes y causando variabilidad en la totalidad de la producción.

Cuando un producto está en la etapa de producción, los problemas que se presenten pueden causar grandes variaciones en la producción, incluso dando lugar a productos con características distintas a lo deseado. En éste caso el control estadístico de procesos (SPC por sus siglas en inglés) permite gestionar la información con la finalidad de detectar la aparición de problemas de forma rápida e identificar las causas que produjeron los problemas, tal que si se lleva a cabo un control continuo dichos problemas pueden ser eliminados. Pero, en procesos sin problemas diagnosticados, también puede haber variabilidad en la producción. Ésta variabilidad es conocida como aleatoria o capacidad del proceso, causadas algunas veces por condiciones ambientales de la fábrica, calidad de la maquinaria o de las materias primas, etc. En dichos casos se debe optar por acciones más costosas que implican el ajuste de las tolerancias de los componentes o subprocesos, modificar el diseño del producto o proceso (lo cual resulta muy difícil cuando se está en producción), inspeccionar el producto saliente para separar los defectuosos, etc.

En la etapa del diseño del proceso se puede lograr que los procesos de producción sean eficientes si se tienen en cuenta las condiciones que pueden causar los problemas, por ejemplo, teniendo en cuenta las condiciones de trabajo que pueden afectar la calidad del producto (temperatura, tipo de maquinas, materias primas, etc) se pueden lograr diseñar procesos menos variables. Entonces, la herramienta que puede ayudar en la optimización de estas condiciones o parámetros es el Diseño de Experimentos.

La fase del diseño del producto es la más importante para atacar la variabilidad que pueden presentar los productos a lo largo de su vida, es decir, desde el momento de fabricación hasta su desuso en el mercado. Tradicionalmente, la manera de estudiar la variabilidad presente en el producto era mediante la recogida de datos mediante un diseño experimental que permitiera realizar un análisis de la varianza (ANOVA) respecto a los factores que habían cambiado: tipo de máquina, proveedor, etc. El problema es que en tales estudios se hace un gran esfuerzo de recogida de datos en relación a los pocos factores que se estudian y casi siempre estos factores son fijados en condiciones habituales de trabajo. Es decir, se intenta descomponer la variabilidad total en sus fuentes para llegar a conocer aquella parte del proceso que contribuye más y por lo tanto hacia dónde se van a dirigir los esfuerzos de mejora. Pero es posible fabricar productos de calidad sin tener que realizar esfuerzos económicos extras si se concibe el producto robusto desde su comienzo; diseñando una metodología de trabajo adecuada y en línea con lo que se conoce como **Ingeniería de Calidad**. Ésta metodología comprende estrategias encaminadas a causar cambios deseables en una situación poco conocida, disponiendo de recursos limitados y utilizando una heurística propia de la disciplina.

La ingeniería de calidad se basa en una filosofía de co-desarrollo, donde todas las partes del proceso se involucran desde el inicio y, de esta manera, los aspectos más críticos del diseño del producto, del diseño del proceso y de la fabricación son atacados desde el comienzo. Es probable que ésta metodología de trabajo consuma más tiempo y recursos que la manera tradicional, donde cada fase trabaja por separado, pero a largo plazo es más rentable ya que se minimizan costes de no-calidad que con la metodología tradicional suelen aparecer en etapas posteriores. En resumen, con la ingeniería de calidad se garantiza un producto robusto desde sus inicios, es decir, no se obtiene por modificaciones de un diseño inicial que responde a otros parámetros.

Así pues, en términos de calidad y costes de producción la etapa inicial del proceso de fabricación (el diseño del producto) es la principal. Por tanto, y en concordancia con los objetivos del presente trabajo, decidimos explorar brevemente las fases que en esta etapa se llevan a cabo, pero lo haremos desde la óptica utilizada por Taguchi, el cual presenta 3 fases y las define como Metodología del diseño robusto (Wu y Wu, 1996).

1.1. El diseño robusto desde la óptica Taguchi

Los métodos de Taguchi para el control de la calidad, en producción en tiempo real (on-line) y fuera de producción no en tiempo real (off-line), representan una propuesta original para reducir la variabilidad del producto. Los métodos on-line comprenden diferentes técnicas para mantener los valores-objetivo y la variación con respecto al objetivo en una planta industrial. En estas técnicas se utilizan cuadros de control estadístico. No obstante, han sido las técnicas del control de calidad off-line las que han distinguido los métodos de Taguchi. El control de calidad

off-line involucra a la función de diseño o de ingeniería de calidad y consiste de tres componentes:

Diseño Primario: Consiste en el diseño conceptual o funcional del producto en respuesta a las necesidades del mercado. El producto debe tener una variación funcional mínima, a causa de factores como el uso. En ese sentido, se emplean diferentes métodos para determinar los requerimientos del consumidor y traducirlos en términos técnicos. Los métodos del *despliegue de la función de calidad* (QFD por sus siglas en inglés) así como la *función de pérdida* se utilizan a menudo en la planificación del sistema.

Diseño Secundario: Comúnmente conocida como diseño de parámetros, es la fase donde se determinan los valores nominales óptimos que garanticen las características de calidad del producto. Implica la identificación de las variables clave del proceso que afectan la variación del producto, y la definición de los niveles parámetro que producirán la menor cantidad de variación en el funcionamiento del producto.

Diseño terciario o diseño de tolerancias: Donde se asegura la variabilidad del producto atacando a la variabilidad de los componentes. Esta etapa resulta costosa por lo que solo se lleva a cabo cuando la variabilidad final del diseño secundario es excesiva

Como resultado del diseño primario se llega a la definición de un producto con una serie de objetivos a cumplir, pero no se tiene todavía el producto en físico. En el diseño secundario se ha de conseguir un prototipo por métodos no convencionales, donde la idea esencial reside en que es posible reducir la variabilidad que generan variables fuera de control al seleccionar valores adecuados en las variables susceptibles de controlarse; de ésta forma se

tiene un producto robusto (insensible) a los cambios inevitables en dichas variables no controladas. Vale la pena acotar que esta fase es determinante en el proceso de producción por lo que, dada su importancia, es necesario crear o provocar escenarios para aprender del comportamiento de las distintas funciones del producto ante diferentes condiciones de los parámetros. Dicha experimentación requiere de una metodología que permita conseguir, de una manera eficiente, la combinación de los niveles de los parámetros que proporcionen un producto cuyas características de calidad estén lo mas cerca posible de los valores nominales definidos, siendo robustos a causas de variabilidad externa, internas y de producción.

El desarrollo del enfoque metodológico de ingeniería de calidad dada por Taguchi, el cual se detalla en la siguiente sección, despertó gran interés en la comunidad estadística occidental en los años 80 dando lugar a discusiones y controversias. Uno de los primeros artículos que aparece en la bibliografía en aquella época intentando aclarar los procedimientos estadísticos empleados por Taguchi se debe a Kackar en 1985 (Maghsoodloo y otros, 2004) y más recientemente Robinson y otros (2004). Dichos artículos presentan un panel de discusión y recogen un sentimiento común que reconoce la importancia de las aportaciones conceptuales y metodológicas de Taguchi para la mejora de la calidad en la industria, pero también una necesidad de mejorar muchos aspectos del método.

Desde la publicación de Kackar han surgido gran cantidad de publicaciones que critican fuertemente las técnicas estadísticas del método Taguchi. De entre las más recientes citamos el trabajo de Maghsoodloo y otros (2004), quienes muestran las fuerzas y limitaciones del enfoque de Taguchi en procesos de ingeniería, centrándose en que sus técnicas no son fáciles de aplicar en la vida real, considerando que las relaciones señal-ruido

(S/N) carecen de rigurosidad estadística para identificar el mejor nivel del factor que minimice las pérdidas de calidad. Por su parte, Tsui (1996) y Montgomery (2001) afirman que el dividir en dos arreglos ortogonales trae como consecuencia el aumento en el número de repeticiones de los experimentos, lo que se traduce en costes innecesarios. Phadke, Fowlkes y Creveling (citados en Su y Chang, 2000) expresan que, en la práctica, el enfoque de Taguchi se limita a elegir el mejor nivel del factor entre un grupo especificado previamente, así como su limitación a parámetros de tipo discreto y el estudio de experimentos para disminuir el rango de los niveles de los factores de control, lo que trae un excesivo número de ensayos y falta de rigor en las conclusiones.

Todo lo anterior ha dado lugar a la proliferación de publicaciones, artículos en revistas especializadas y cursos enfocados en propuestas estadísticas para la mejora del diseño robusto; lo que prueba el gran interés que ha tenido el tema no sólo en el ámbito industrial sino en el académico. Se podría decir entonces que la metodología propuesta inicialmente por Taguchi ha evolucionado, pero aun siguen existiendo puntos críticos en los cuales no se opta por una postura global. Entre esos puntos se pueden mencionar las métricas a utilizar, el análisis de la información producto de la experimentación y la selección de las condiciones óptimas. Todo ello se encierra en lo que se conoce como *el problema del diseño de parámetros*.

Entre los trabajos más recientes hallados en la literatura están las propuestas de Su y Chang (2000), en el caso univariante y Chang (2006, 2008), en el caso multivariante. Tales trabajos basan sus aportes en la combinación de las Redes Neuronales Artificiales (ANN) con Simulated Annealing (SA). Por su parte Chang (2005) y Chang y Chen (2011), en casos univariantes y multivariantes respectivamente, emplean las ANN

pero ahora en combinación con Algoritmos Genéticos (GA). Las ANN constituyen una poderosa herramienta en procesos predictivos pero en la mayoría de los casos resultan inestables estadísticamente, no obstante, consideramos que combinar técnicas de minería de datos con técnicas de optimización heurística puede seguir dando buenos resultados y por tal razón nuestra propuesta metodológica se enmarca en combinar tales herramientas con el fin de proponer una metodología completa y robusta para la mejora del diseño de parámetros.

1.2. Objetivos

1.2.1. Objetivo general

Diseñar una metodología estadística, completa y robusta para la mejora del diseño de parámetros que combine estrategias basadas en árboles de regresión (CART) y Random Forest (RF) con Algoritmos Genéticos (GA), tanto para el caso univariante como para el caso multivariante.

1.2.2. Objetivos específicos

1. Diseñar funciones de normalización que permitan minimizar la variabilidad propia del conjunto de datos.
2. Diseñar estrategias basadas en CART y RF para la modelización del conjunto de parámetros determinados.
3. Comparar los rendimientos de las estrategias del objetivo anterior con estrategias basadas en ANN, propuestas por distintos autores, con el fin de determinar el mejor esquema de modelización univariante en términos de robustez y estabilidad estadística.
4. Generalizar los resultados del objetivo 2 e incorporar las métricas acordes para el diseño de la estrategia de modelización en casos multivariantes.
5. Comparar los rendimientos de las estrategias del objetivo anterior con estrategias basadas en ANN, propuestas por distintos autores, con el fin de determinar el mejor esquema de modelización multivariante en términos de robustez y estabilidad estadística.
6. Diseñar el esquema de optimización de la función objetivo como un híbrido entre la estrategia definitiva de modelización univariante y

los GA.

7. Generalizar los resultados del objetivo anterior e incorporar las métricas acordes para el diseño del esquema de optimización de la función objetivo como un híbrido entre la estrategia definitiva de modelización multivariante y los GA.
8. Comparar los resultados de nuestras propuestas metodológicas con los obtenidos por otros autores que han tratado el tema.

1.3. Fases para el desarrollo de la metodología propuesta

Nuestra propuesta metodológica se basa en la combinación de funciones de normalización con técnicas de minería de datos y técnicas de optimización heurística, por tal razón se desarrollará mediante las siguientes fases:

Normalización: En primera instancia, la normalización permite reducir la variabilidad propia del conjunto de datos, además nuestra metodología pretende ser una alternativa en el uso de las ANN como técnica para el establecimiento de la relación input/outputs, por lo que a fines de comparaciones numéricas es vital la igualdad de escalas. Esta fase tiene como finalidad seleccionar la mejor función de normalización (en términos de variabilidad) de entre tres propuestas diferentes diseñadas previamente.

Modelización: Corresponde a un esquema de trabajo cuyo producto final será la determinación de la función objetivo para la predicción de la respuesta de un conjunto de parámetros, en vista de no contar con un modelo a priori. A tal fin se establecen algoritmos que combinan técnicas de remuestreo y minería de datos, las cuales hemos bautizado

bajo el nombre de *estrategias basadas en árboles de regresión (CART) y random forest (RF)* en contraste con las basadas en ANN. Como estadístico de prueba para la comparación de las distintas estrategias se establece la raíz del error cuadrático medio (RMSE) tal que, se elige como algoritmo de modelización el que proporcione el menor RMSE bajo la mejor función de normalización de la fase anterior.

Optimización: Para finalizar la metodología propuesta, se desarrolla la fase de optimización, donde el objetivo es la determinación de los niveles de los parámetros que proporcione el mejor valor de la variable respuesta, de acuerdo con la característica de calidad predeterminada. A tal fin, se construye un GA que incorpore las propiedades del esquema de modelización seleccionado en la fase anterior en el diseño de los correspondientes operadores genéticos.

Vale la pena acotar que, para el establecimiento de la metodología propuesta se desarrolló inicialmente el caso univariante y se validaron los resultados mediante dos conjunto de datos reales. Posteriormente, bajo el apoyo de los resultados obtenidos, se establece la metodología al caso multivariante y se validaron los resultados mediante dos casos de ilustración diferentes. Todo lo anterior queda reflejado en la figura 1.1. Sin embargo para una mejor comprensión de la metodología propuesta, estructuramos el presente trabajo tal que se organizan los casos univariante y multivariante en correspondencia con las fases de modelización y optimización, como se explicará en la siguiente sección.

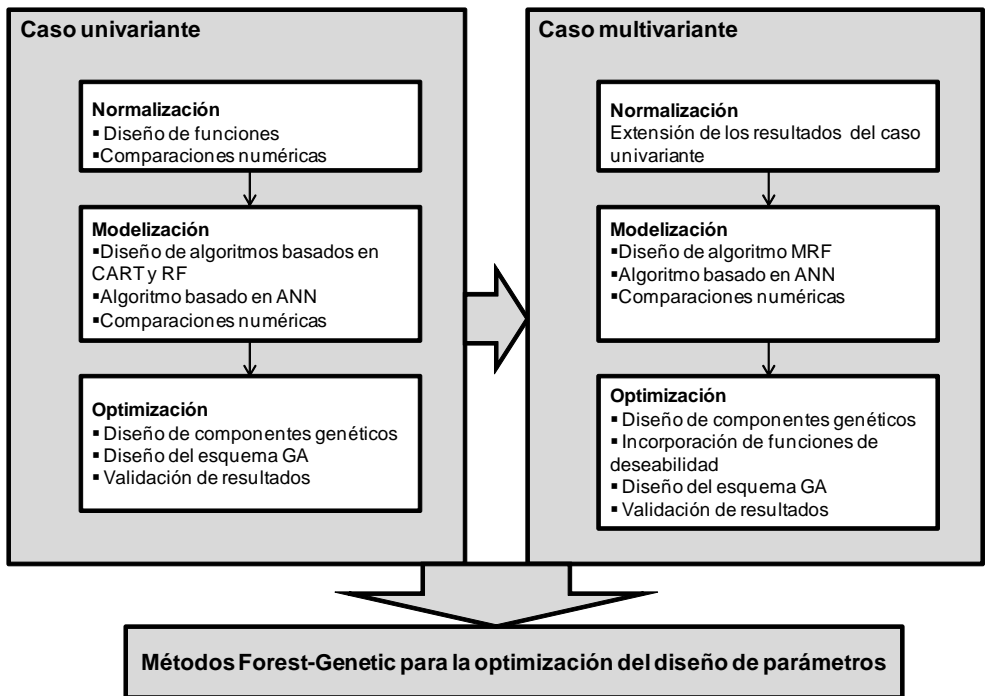


Figura 1.1: Fases para el desarrollo de la metodología propuesta

1.4. Organización del trabajo

Para el logro de los objetivos del presente trabajo hemos distribuido el manuscrito por partes, las cuales a su vez se han dividido en capítulos con objetivos y aportes bien marcados. Así pues, iniciamos el capítulo 1 con una presentación de la metodología del diseño robusto desde la óptica Taguchi, la cual da lugar a la motivación del presente trabajo y el esclarecimiento de los objetivos perseguidos. Seguidamente, el capítulo 2 presenta el estado del arte, donde no se pretende realizar una recopilación exhaustiva, sino mas bien presentar una cronología de los últimos aportes hallados en la literatura dirigidos a la mejora del diseño de parámetros, y presentamos en pleno la metodología propuesta por Taguchi.

La segunda parte del presente trabajo corresponde al desarrollo de las fases de normalización y modelización tanto para el caso univariante como para el caso multivariante. Proponemos estrategias basadas en CART y RF en el caso univariante y generalizamos los resultados obtenidos para el diseño del esquema de modelización multivariante. Además, para ambos casos, se incorporan las ANN con fines comparativos. Todo lo anterior se estructura como sigue:

- El capítulo 3 presenta, de forma introductoria, el basamento teórico que nos permitió el desarrollo y selección de las estrategias de modelización con el fin de evitar vacíos conceptuales.
- El capítulo 4 comprende la propuesta de preparación previa del conjunto de datos mediante funciones de normalización, el ajuste de los árboles a desarrollar dentro de cada algoritmo de remuestreo así como los parámetros de control de CART y RF. Además, se presentan en detalle el diseño de las estrategias para la modelización univariante.
- El capítulo 5 se presenta como una generalización del capítulo anterior al caso multivariante. Incluimos la fase de normalización y proponemos el estudio de diferentes métricas de modelización para la integración de las múltiples características de calidad en secuencias de árboles de regresión univariante. Lo anterior tendrá como producto final las estimaciones de dichas respuestas y por tanto la determinación de la función objetivo del conjunto de parámetros bajo estudio. A dicha propuesta la hemos bautizado bajo el nombre de *Random Forest Multivariante*.
- En el capítulo 6 presentamos, a manera de conclusión de la presente

fase, el esquema final de modelización univariante y multivariante que va a ser empleado en las fases de optimización correspondientes, dando respuesta a los 3 primeros objetivos específicos planteados.

Vale la pena mencionar que tanto el capítulo 4 como el 5 están acompañados con los respectivos casos de ilustración que permiten validar nuestros resultados y establecer comparaciones numéricas con otros trabajos hallados en la literatura en torno a la problemática tratada.

La tercera parte del presente trabajo desarrolla la fase de optimización como punto de cierre a nuestra propuesta metodológica para la mejora del diseño de parámetros. A tal fin consideramos pertinente presentar los capítulos como sigue:

- El capítulo 7 presenta una breve introducción teórica de GA como la metaheurística en la cual se orienta nuestra propuesta de optimización y las funciones de deseabilidad como elemento añadido para el caso multivariante.
- El capítulo 8 presenta el diseño de nuestra propuesta de optimización en el caso univariante. Tal propuesta se fundamenta en GA, donde se incorporan las propiedades del esquema definitivo de modelización presentado en el capítulo 6 para el caso univariante en los operadores genéticos. En el capítulo 9 se generalizan los resultados obtenidos en el capítulo anterior y se incorporan las funciones de deseabilidad para el diseño de los operadores genéticos en la estrategia de optimización multivariante. Además, tanto en el capítulo 8 como en el 9, validamos y comparamos numéricamente la metodología propuesta mediante los casos de ilustración considerados en los capítulos 4 y 5 respectivamente.

- En el capítulo 10 presentamos, a manera de conclusión de la presente fase, el esquema final de optimización tanto univariante como multivariante, dando respuesta a los objetivos específicos 6 y 7.

Finalizamos con la cuarta parte, dedicada a conclusiones y futuras líneas de investigación. En el capítulo 11 presentamos el esquema final de toda nuestra propuesta metodológica al problema del diseño de parámetros así como un resumen del cumplimiento de los objetivos específicos. Hemos unido las fases de normalización, modelización y optimización, tanto para el caso univariante como para el caso multivariante con la finalidad de presentar al lector nuestra propuesta de forma completa. Tales propuestas metodológicas las hemos llamado respectivamente Forest-Genetic univariante y Forest-Genetic multivariante además, se presentan las principales virtudes que tales métodos poseen.

Capítulo 2

El estado del arte

Las ideas de Taguchi constituyen grandes contribuciones a la ingeniería de calidad, sin embargo desde sus inicios existían dudas de que la propuesta realmente diera lugar a productos o procesos realmente robustos y con buen comportamiento funcional. Occidentales destacados como Box y Meyer (1986), Box y otros (1988) y Ryan (1989) calificaban los procedimientos estadísticos de Taguchi como innecesariamente complicados e ineficientes, mostrando en sus obras revisiones detalladas a las limitaciones de las relaciones señal-ruido (S/N, por sus siglas en inglés), la forma en que se lleva a cabo la experimentación, la optimización, etc. Se aceptaba que para medir la calidad de un producto era necesaria una función de pérdidas que tuviera en cuenta las variaciones funcionales que pudiera experimentar el comportamiento del producto en manos del cliente, pero no quedaba claro los procedimientos estadísticos empleados por Taguchi para la selección de la combinación de los parámetros al considerar las relaciones S/N con poca rigurosidad, así como la idea de realizar experimentos confirmatorios que se traduce en costes innecesarios y la terminología poco usual que es empleada.

Estos y otros aspectos han dado lugar a numerosos estudios que intentan mejorar diferentes aspectos del método; de hecho, el presente trabajo constituye un aporte al diseño de parámetros bajo el contexto de la metodología Taguchi, por lo que hemos decidido presentar las pautas fundamentales del método Taguchi en el presente capítulo y, seguidamente, en la sección que hemos definido como la evolución del método Taguchi, presentamos los aportes más recientes encontrados en la literatura en pro de su mejora.

2.1. Metodología Taguchi

Según el enfoque Taguchi, la calidad de un producto decrece en la medida en que los valores de sus características funcionales se alejan de los valores objetivos (target), y debe medirse en función de la pérdida que sufre la sociedad como producto de una mala calidad. De aquí el nombre de *Función de Pérdida*, definida como $L = k(x-T)^2$, donde k es una constante específica de cada caso considerado, T es un valor objetivo (T mide la calidad nominal), x es el valor de la característica cuantitativa de la calidad considerada y finalmente L es la pérdida. Entonces, si x es una variable aleatoria de media μ y desviación típica σ , la pérdida media es:

$$E(L) = k[\sigma^2 + (\mu - T)^2] \quad (2.1)$$

Así pues, Taguchi se apartó de la idea convencional basada en producir dentro de los márgenes de tolerancia, postulando que el costo de la mala calidad se incrementa cuadráticamente con el alejamiento del valor objetivo y se acerca a cero cuando el desvío con respecto al valor objetivo es nulo. La figura 2.1 muestra la función de pérdida de dos procesos A y B, donde LS y US representan los márgenes de tolerancia. Se puede notar cómo en el

proceso de B disminuye la función de pérdida mediante la reducción de la variabilidad con respecto al valor objetivo, en comparación con el proceso A. Esta variabilidad es el principal enemigo de la calidad y el momento más adecuado para combatirla, sin incrementos de costo, es en la fases tempranas del diseño.

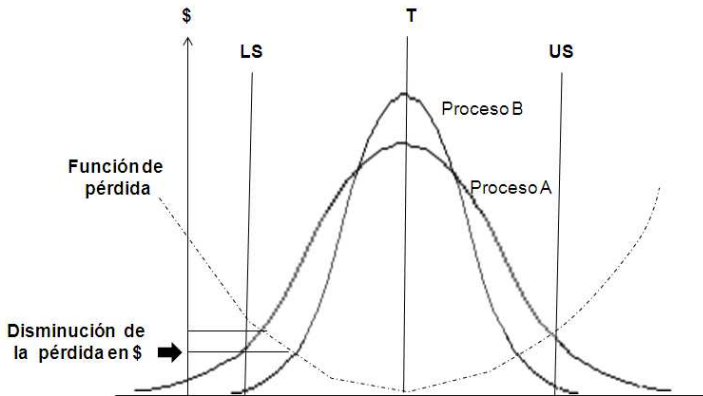


Figura 2.1: Función de pérdida de Taguchi

Taguchi emplea el diseño de parámetros para incrementar la calidad de nuevos productos y procesos, donde la idea básica es identificar la robustez del sistema ante variaciones incontrolables (ruido) mediante la exploración de los niveles apropiados de los factores de control. Por tal razón, el enfoque Taguchi es comúnmente conocido como *diseño robusto de parámetros*, donde la palabra *diseño* se refiere más al diseño de sistemas que al diseño estadístico de experimentos. El diseño estadístico de experimentos es el medio para llegar al diseño robusto de parámetros. Para Taguchi, el error no se maneja como un efecto aleatorio para estudiar estadísticamente el efecto factorial, sino que en él se incluyen todas las fuentes de variación, alteraciones del entorno, deterioro y variabilidad de los productos. Por tanto, éstos factores son intencionalmente incluidos con el propósito de

introducir ruido en los experimentos. Otro aspecto a resaltar es que el diseño de experimentos tradicional parte de dos supuestos fundamentales: normalidad de los errores y homogeneidad de varianzas; la metodología Taguchi no espera que los factores de ruido sigan tal patrón de comportamiento, pues el propósito es descubrir si existen diferencias entre las magnitudes de los errores de los niveles de un factor control. Entonces, si existe tal diferencia, se puede elegir el nivel del factor de control que posea el menor error determinando así la condición más robusta, pero si hay igualdad de varianzas el propósito de minimizar el error se torna imposible.

Para el análisis del diseño de parámetros, Taguchi define 3 tipos de factores: factor señal, factor ruido y factor control. Los factores señal determinan el rango de configuraciones a ser considerados en el diseño robusto, afectando directamente a la respuesta media del proceso pero sin alterar la variabilidad; los factores de ruido son parámetros que no pueden ser controlados por el diseño y constituyen la fuente de variabilidad del sistema. Finalmente, los factores control son los parámetros especificados por el diseño y que deben ser optimizados con el fin de hacer menos sensibles las respuestas a los factores de ruido (Maghsoodloo y otros, 2004). Dicha clasificación de los factores constituye el primer paso de la metodología Taguchi, la cual podemos resumir como sigue.

- Selección de la característica de calidad a medir e identificación de los factores control, señal y ruido.
- Experimentación según el diseño determinado, que suelen ser matrices ortogonales (Orthogonal Arrays) de la forma L_{18} y L_{27} (Wu y Wu, 1996).
- Selección de la medida S/N.

- Minimización de la función de pérdida.
- Realización de experimentos confirmatorios.

A continuación se presenta de una manera más amplia el procedimiento a seguir en cada paso, pero acotamos que el procedimiento de la selección de la medida S/N la hemos dejado al final del presente capítulo dada la importancia que tiene a efectos del presente trabajo.

Selección de la característica de calidad a medir

Dada una función de interés, se debe determinar entre las posibles características cuantificables asociadas a dicha función, la característica (o vector de características en el caso multivariante) con la que se trabajará durante la fase de experimentación, la cual se denota como Y . En general, se tiende a tomar características afectadas de forma lineal por los factores a estudio; por lo tanto, se tiene en cuenta el valor esperado de la función, su localización y, en menor medida, la dispersión que dicha característica presenta debido a los factores de ruido. Como veremos en la segunda parte del presente trabajo, dedicado a la modelización, la verdadera relación entre la característica Y y los factores de estudio afectan al éxito de la metodología del diseño robusto.

Una vez determinados los factores control (X), señal (M) y ruido (Z), la relación con Y puede ser expresada como $Y = f(M, X, Z) + \varepsilon$, donde el objetivo es encontrar las condiciones de los factores control que den lugar a un valor mínimo de la función de pérdidas, es decir, procurar que Y esté lo más cerca posible de su valor óptimo y permanezca insensible a las variaciones de los factores de ruido.

Selección y análisis de la matriz de diseño

Taguchi recomienda realizar la experimentación bajo una matriz de diseño compuesta en dos partes: una matriz interna y otra externa. Los factores de control son asignados a la matriz interna y los factores señal y ruido a la matriz externa, tal como lo ilustra la tabla 2.1. Wu y Wu (1996), recomiendan el uso de las matrices ortogonales L_{12} , L_{18} y L_{36} en el diseño interno: L_{12} para factores a dos niveles, L_{18} para factores a tres niveles y L_{36} para los casos de simulación. En el caso del diseño externo, recomienda la L_4 o L_9 para reunir los factores de ruido y señal, aunque sugiere componer los factores de ruido en uno solo y combinarlo con los factores señal. La tabla 2.1, muestra un esquema de los diseños internos y externos, considerando una matriz L_{18} y L_4 respectivamente, donde $X = \{A, B, C, D, E, F, G\}$ son los factores de control (A con dos niveles y el resto con 3 niveles), $Z = \{N_1, N_2\}$ y $M = \{M1, M2\}$ son los factores de ruido y señal respectivamente, a dos niveles cada uno.

Taguchi recomienda evaluar en el diseño interior la respuesta promedio de cada repetición del experimento y analizar la variación (ANOVA) de los resultados con la relación S/N apropiada. Así, para el caso de la tabla 2.1 se obtendrán 18 relaciones S/N, las correspondientes medias y desviaciones típicas, como se muestra en la tabla 2.2 donde $i = 1, \dots, 18$. Luego se calculan los subtotales por cada nivel de los factores control, sus sumas de cuadrados (SC) y cuadrados medios (CM), asignando los *Ranks* en orden decreciente a partir del mayor CM(S/N). Se consideran óptimos los niveles de los factores que maximicen el S/N y en caso de factores no significativos se elige aquel que, por conocimientos previos, se considere el más económico. Como se explicará en la siguiente sección, el S/N difiere según el objetivo del experimento sea reducir la variabilidad alrededor de un valor objetivo fijado, producir un efecto resultante lo mayor posible, o inversamente, producir el menor efecto posible.

Tabla 2.1: Matriz de diseño compuesta según la metodología Taguchi

Nº	L_{18}								L_4			
	A	B	C	D	E	F	G	H	M1		M2	
									N_1	N_2	N_1	N_2
1	1	1	1	1	1	1	1	1				
2	1	1	2	2	2	2	2	2				
3	1	1	3	3	3	3	3	3				
4	1	2	1	1	2	2	3	3				
5	1	2	2	2	3	3	1	1				
6	1	2	3	3	1	1	2	2				
7	1	3	1	2	1	3	2	3				
8	1	3	2	3	2	1	3	1				
9	1	3	3	1	3	2	1	2				
10	2	1	1	3	3	2	2	1				
11	2	1	2	1	1	3	3	2				
12	2	1	3	2	2	1	1	3				
13	2	2	1	2	3	1	3	2				
14	2	2	2	3	1	2	1	3				
15	2	2	3	1	2	3	2	1				
16	2	3	1	3	2	3	1	2				
17	2	3	2	1	3	1	2	3				
18	2	3	3	2	1	2	3	1				

Minimización de la función de pérdida

Wu y Wu (1996) en su obra aclaran que en casos de problemas multirespuesta es necesario definir previamente una función de pérdidas para valorar conjuntamente dichas respuestas pero, en general, la selección de las condiciones óptimas en el método Taguchi se pueden resumir en los siguientes dos pasos.

1. Seleccionar los niveles de los factores de control que maximicen la medida S/N, ignorando la media.
2. Manteniendo los niveles de los factores fijados en el paso 1, seleccionar los niveles de los factores señal que lleven a la media al valor deseado.

Realización de experimentos confirmatorios

Antes de dar el resultado como definitivo, Taguchi propone la realización de una serie de experimentos para confirmar que en las condiciones seleccionadas el comportamiento de Y es el esperado. En caso de tener grandes discrepancias se debería de volver hacia atrás en búsqueda de posibles errores de interpretación o análisis, revisando la propia selección de factores de control, ruido y señal si fuera necesario.

2.1.1. Relaciones S/N

La relación señal-ruido (S/N) es un índice que, históricamente, ha sido utilizado en la industria de la comunicación para evaluar los sistemas de comunicación. En ingeniería de calidad, el concepto de S/N ha sido adoptado por Taguchi para evaluar, en la etapa de diseño de parámetros, la calidad de un producto o un proceso teniendo en cuenta los dos conceptos: la señal y el ruido (Wu y Wu, 1996), de esta manera, si cada fila i ($i = 1, \dots, 18$) de la tabla 2.2 es considerada como un prototipo diferente, η_i da una medida

de calidad de cada prototipo y es estimada a partir de las observaciones y_{ij} a lo largo de las condiciones de ruido.

Tabla 2.2: Esquema de análisis según la metodología Taguchi

Nº	Media	Desv. Típica	S/N
1			
2			
3			
4			
.	\hat{y}_i	S_i	η_i
.			
.			
18			

Tradicionalmente, las mejoras de calidad en un sistema se consiguen localizando las causas de variación dentro de las contribuciones de repetitividad, reproducibilidad y estabilidad. Una vez identificadas las fuentes de variación, se realizan esfuerzos para eliminarlas o reducirlas. En ingeniería de calidad, esta actividad es considerada como diseño de tolerancias ya que trata con los factores de ruido, por tanto, el diseño de parámetros debe ser previo al diseño de tolerancias puesto que los efectos de ruido pueden reducirse de forma más económica. Otros aspectos importantes son la sensibilidad y linealidad; la sensibilidad es la capacidad de distinguir las diferencias entre los objetos (muestras) a medir. La linealidad se refiere a las posibles diferencias en el sesgo de las medidas obtenidas en función de la magnitud de medida (por ejemplo: pieza pequeña, sesgo pequeño, pieza grande, sesgo grande). Entonces, en un sistema, se estudia la relación entre

la entrada y la salida, donde el valor verdadero del objeto a medir es la entrada, y el resultado de la medida es la salida. Ahora bien, si el valor verdadero del objeto no se encuentra limitado por un único valor sino por un rango de posibles valores verdaderos, entonces se hablará de *sistemas de medida*.

Un buen sistema de medida debe tener sensibilidad a diferentes entradas (diferentes objetos a medir), debe ser fácil de calibrar y tener una pequeña variabilidad (Wu y Wu, 1996). Cuando se utiliza la relación S/N para evaluar un sistema de medida, estos tres elementos se combinan como se muestra en (2.2).

$$\begin{aligned} S/N &= \frac{\text{Potencia de la señal}}{\text{Potencia del ruido}} \\ &= \frac{(\text{Sensibilidad por unidad de entrada})^2}{(\text{Error por medida})^2} \end{aligned} \quad (2.2)$$

En efecto, relacionando S/N con los aspectos de sensibilidad, variabilidad y linealidad tenemos que, cuando el sistema de medida tiene sensibilidad entre muestras, la relación S/N crece y cuando el error de medida aumenta, S/N decrece. Finalmente, la facilidad de calibración, se consigue con la proporcionalidad de la relación entrada-salida (relación lineal). Cuanto más lineal es dicha relación, mayor será la relación S/N (Joseph y Wu, 2002; Miller y Wu, 1996; Zang y Mottershead, 2005). Acotamos que, como se verá con más detalle en el siguiente capítulo, esta es una de las debilidades del enfoque Taguchi, pues habitualmente las relaciones entrada-salida no suelen ser de tipo lineal.

Por otra parte, Taguchi define dos tipos de respuesta, dependiendo de la característica de calidad en interés: *característica estática* y *característica*

dinámica. La característica estática o *no dinámica* (Wu y Wu, 1996) se refiere a valores objetivos fijados previamente; éstos incluyen el cero, el infinito o valores nominales particulares y se evalúan con las relaciones S/N con las categorías: *menor es mejor*, *mayor es mejor*, *nominal es mejor* (STB, LTB y NTB respectivamente). Cuando el objetivo no está limitado a medir la muestra teniendo sólo un valor verdadero, es decir, el sistema de medida esta diseñado para medir las muestras teniendo un rango de valores verdaderos, entonces el error debe ser pequeño para todo el rango de medida. Aquí nace la razón de los S/N dinámicos. Las relaciones de tipo dinámico para Taguchi, son empleadas para mejorar la robustez de la función de un producto en el rango de salida, mientras que las no dinámicas también mejoran la robustez pero para cierto valor objetivo de salida.

Taguchi recomienda que, aun cuando se considere el planteamiento definido como *no dinámico*, se debe plantear primeramente la optimización de la función ideal (objetivo), ya que cada sistema de medida tiene un propósito y para cada propósito, existe una función que transforma una entrada en una salida y éste es el punto más importante en el rendimiento del propio sistema. Sin embargo, a efectos del presente trabajo, a continuación se detalla un poco más la metodología Taguchi aplicada a sistemas dinámicos y no dinámicos.

2.1.2. Metodología Taguchi en sistemas dinámicos

En sistemas dinámicos, si un producto o proceso que va a ser optimizado tiene un factor señal (inputs) que incide directamente sobre las respuestas (outputs), como se puede observar en la figura 2.2, entonces la optimización consiste en determinar los mejores niveles del factor control tal que las respuestas se acerquen lo más posible a los objetivos del sistema (target).

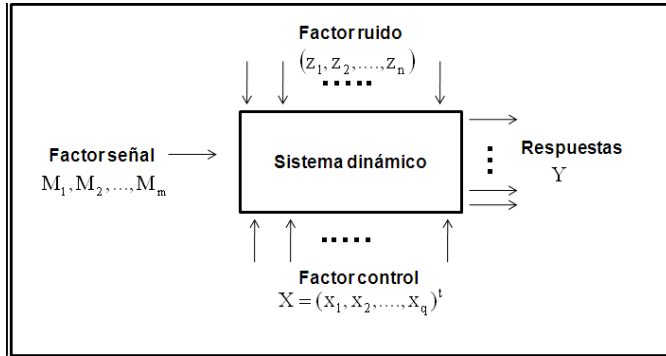


Figura 2.2: Sistemas dinámicos

Los sistemas dinámicos pueden dividirse según el tipo de variable en cuatro tipos (Wu y Yeh, 2005), los cuales se describen brevemente, a efectos de los objetivos planteados en el presente trabajo

- Continuas-Continuas
- Continuas-Digital
- Digital-Continuas
- Digital-Digital

Continuas-Continuas (C-C):

En este tipo de sistema, se considera la característica de calidad Y como una función del factor señal M . Tanto Y como M con dominio en \mathbb{R} . Las respuestas de Y (outputs) son afectados directamente por el conjunto de señal, que denotaremos como $M = \{M_1, M_2, \dots, M_m\}$ y para cada valor señal se considera el conjunto $Z = \{z_1, z_2, \dots, z_n\}$ como ruido. Entonces, si se denota a y_{ij} como la observación para el q -ésimo conjunto del factor control $\mathbf{X} = (x_1, x_2, \dots, x_q)$ con factor señal M_i y ruido z_j , la función ideal puede ser expresada como $y_{ij} = \beta M_i + \varepsilon_{ij}$, donde β representa la sensibilidad del sistema y ε el error aleatorio.

Estimando el factor β por el método de mínimos cuadrados tenemos:

$$SEE = \sum_{i=1}^m \sum_{j=1}^m \varepsilon_{ij}^2 = \sum_{i=1}^m \sum_{j=1}^m (y_{ij} - \beta M_i)^2$$

derivando SEE con respecto a β

$$\frac{d}{d\beta} \left[\sum_{i=1}^m \sum_{j=1}^m (y_{ij} - \beta M_i)^2 \right] = 2 \sum_{i=1}^m \sum_{j=1}^m (y_{ij} - \beta M_i) M_i \quad (2.3)$$

igualando la ecuación (2.3) a cero, se obtiene

$$\beta = \frac{\sum_{i=1}^m \sum_{j=1}^n y_{ij}}{M_i} \quad (2.4)$$

Wu y Yeh (2005) expresan la pérdida de calidad promedio después del ajuste como:

$$\begin{aligned} Q_a(\mathbf{X}) &= \frac{k}{mn} \sum_{i=1}^m \sum_{j=1}^n (v_{ij} - M_i)^2 \\ &= \frac{k}{mn} \sum_{i=1}^m \sum_{j=1}^n \left(\frac{y_{ij}}{\beta} - M_i \right)^2 = \frac{k}{mn} \sum_{i=1}^m \sum_{j=1}^n \frac{(y_{ij} - \beta M_i)^2}{\beta^2} \\ &= \frac{k}{\beta^2} \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (y_{ij} - \beta M_i)^2 \end{aligned}$$

donde k es la constante que representa el coeficiente de pérdida.

Como

$$\sigma_\varepsilon^2 = \frac{1}{mn - 1} \sum_{i=1}^m \sum_{j=1}^n (y_{ij} - \beta M_i)^2 \Rightarrow \sum_{i=1}^m \sum_{j=1}^n (y_{ij} - \beta M_i)^2 = \sigma_\varepsilon^2 (mn - 1)$$

Entonces:

$$Q_a(\mathbf{X}) = \frac{k}{\beta^2} \frac{(mn-1)\sigma_\varepsilon^2}{mn} \Rightarrow Q_a(\mathbf{X}) \cong \frac{k\sigma_\varepsilon^2}{\beta^2} \quad (2.5)$$

Donde $v_{ij} = \frac{y_{ij}}{\beta}$ es el valor ajustado, y $\sigma_\varepsilon^2 = \frac{1}{mn-1} \sum_{i=1}^m \sum_{j=1}^n (y_{ij} - \beta M_i)^2$ es la varianza del error.

De todo lo anterior, Taguchi (Wu y Wu, 1996) define el S/N para el tipo C-C como :

$$\eta = -10 \log Q_a(\mathbf{X}) = 10 \log \frac{\beta^2}{\sigma_\varepsilon^2} \quad (2.6)$$

Como se puede ver en (2.5), la constante k en $Q_a(\mathbf{X})$ es ignorada, ya que no tiene efecto en la optimización de una sola característica de calidad. En experimentos con múltiples características de calidad, como se verá más adelante, k juega un papel importante para el ajuste entre dichas características. Por otra parte, considerando la ecuación (2.6), minimizar Q_a es equivalente a maximizar S/N (Wu y Yeh, 2005). Sin embargo, Taguchi sugiere, en este caso, la optimización en dos pasos de forma similar al caso estático *nominal es mejor* (Wu y Wu, 1996). En efecto, Taguchi para caso estático *nominal es mejor*, sugiere primero maximizar S/N para conseguir la mayor robustez, luego ajustar la media (pendiente β en el caso dinámico) al valor deseado mediante el factor adecuado. Así, cualquier factor de control que tiene efecto sobre la media, pero no sobre η puede servir como factor de escala.

Continuas-Digital (C-D):

En este tipo de sistema, el conjunto de entrada (input) es de tipo continuo y el de salida (outputs) es discreto. Entonces, si el conjunto de salida posee s niveles, el problema puede analizarse dividiendo dicho conjunto en un conjunto para cada nivel y posteriormente pueden ser vistos como problemas de tipo dinámicos C-C o estáticos *nominal es mejor* (Wu y Yeh, 2005).

Digital-Continuas (D-C):

Este tipo de sistema se comporta de manera similar al caso C-D, pero en este caso, es el conjunto de entrada el que se puede dividir en tantos conjuntos como niveles tenga el conjunto de entrada y proceder como problemas de tipo estático *nominal es mejor*.

Digital-Digital (D-D):

En este tipo de sistema, tanto el conjunto de entrada como el conjunto de salida son de tipo discreto y el factor señal tiene los niveles $M_0 = 0$ y $M_1 = 1$. Entonces, la función ideal desearía que cada vez que la señal cero (0) es transmitida, ésta debería ser recibida como cero y cada vez que la señal 1 es transmitida, la misma debería ser recibida como 1. Entonces, se define a p como la probabilidad de recibir 1 cuando 0 es transmitido, y q la probabilidad de recibir 0 cuando 1 es transmitido. Todo ello bajo un determinado conjunto de factores control y condiciones de ruido. Para que tal transferencia sea lograda de forma más eficiente, Taguchi recomienda cambiar el valor del umbral y hacer $p = q$, entonces propone hacer

$$p' = \left[1 + \sqrt{\frac{1-p}{p} \cdot \frac{1-q}{q}} \right]^{-1}$$

tal que $p' = q'$. La pendiente de la señal transmitida $\beta = 1 - 2p'$ debe pasar

por el punto (0.5,0.5) y así la pérdida después del ajuste de la pendiente estará dada por

$$Q_a(\mathbf{X}) = k \cdot \frac{\sigma_\varepsilon^2}{\beta^2} = k \cdot \frac{p'(1-p')}{(1-2p')^2}$$

y el S/N como $\eta = -10 \log Q_a(\mathbf{X})$. Por tanto, la estrategia es maximizar η y usar el factor de control, que no afecta a η , pero que pueda alterar el ratio $p \setminus q$ para igualar los errores de probabilidad.

Finalmente y considerando que los objetivos del presente trabajo también abarcan casos de múltiples respuestas, presentamos brevemente el enfoque que Taguchi da a su metodología en dichos casos.

2.1.3. Metodología Taguchi en sistemas dinámicos con múltiples respuestas

En un sistema dinámico con múltiples respuestas, también conocido como *Sistemas dinámicos multi-target* (Miller y Wu, 1996), el factor señal es usado para ajustar el funcionamiento del sistema acorde a diferentes valores objetivos (target). En general, un sistema con múltiples respuestas puede ser visto como una extensión del sistema dinámico tratado en el apartado anterior, por tal razón y a efectos del presente trabajo presentamos dicha extensión para el tipo continuo-continuo (C-C).

Supongamos un sistema dinámico C-C con características de calidad Y_1, \dots, Y_p , entonces las respuestas de Y_i , son directamente afectados por el factor señal M_1, M_2, \dots, M_m y las condiciones de ruido (z_1, z_2, \dots, z_n) . Por lo tanto, definimos y_{ijk} como la observación de la característica de calidad Y_i en el q -ésimo factor control $\mathbf{X} = (x_1, x_2, \dots, x_q)$ cuando el factor señal es M_j y z_k el factor ruido. Ajustando todo lo anterior a la figura 2.2

se obtiene la figura 2.3

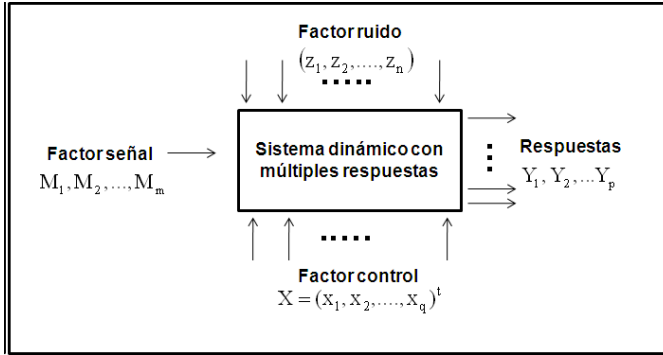


Figura 2.3: Sistemas dinámicos con múltiples respuestas

Wu y Yeh (2005) expresan el promedio de pérdida de calidad como sigue:

$$\begin{aligned}
 Q_a(\mathbf{X}) &= \sum_{i=1}^p \frac{k_i}{\beta_i^2} \frac{1}{mn} \sum_{k=1}^m \sum_{l=1}^n (y_{ijkl} - \beta_i M_k)^2 \\
 &+ \sum_{i=1}^p \sum_{i < j}^p \frac{k_{ij}}{\beta_i \beta_j} \frac{1}{mn} \sum_{k=1}^m \sum_{l=1}^n (y_{ikl} - \beta_i M_k) (y_{jkl} - \beta_j M_k) \\
 &= \sum_{i=1}^p \frac{k_i}{\beta_i^2} \frac{1}{mn} \sum_{k=1}^m \sum_{l=1}^n \varepsilon_{ijkl}^2 + \sum_{i=1}^p \sum_{i < j}^p \frac{k_{ij}}{\beta_i \beta_j} \frac{1}{mn} \sum_{k=1}^m \sum_{l=1}^n \varepsilon_{ikl}^2 \varepsilon_{jkl}^2 \\
 &= \sum_{i=1}^p \frac{k_i}{\beta_i^2} \sigma_{i\varepsilon}^2 + \sum_{i=1}^p \sum_{i < j}^p \frac{k_{ij}}{\beta_i \beta_j} (\sigma_{Y_i Y_j} - \sigma_{Y_i^* Y_j^*})
 \end{aligned}$$

donde k_i es el coeficiente de pérdida de Y_i , k_{ij} es el coeficiente de pérdida correlacionada de Y_i y Y_j , $\sigma_{Y_i Y_j}$ es la covarianza de las observaciones Y_i y Y_j , y $\sigma_{Y_i^* Y_j^*}$ la covarianza de los valores predictivos de Y_i y Y_j .

Así pues, la función objetivo resulta: $Y_i^* = \beta_i M_k$. El intervalo de k_{ij}

está restringido a:

$$|k_{ij}| \leq \left| \frac{\beta_i \beta_j}{\sigma_{ij} - \sigma_{i^*j^*}} \right| \cdot \text{Min} \left(k_i \frac{\sigma_{i\varepsilon}^2}{\beta_i^2}, k_i \frac{\sigma_{i\varepsilon}^2}{\beta_i^2} \right)$$

El objetivo de la optimización de parámetros es encontrar el conjunto de factores de control X que minimice a $Q_a(\mathbf{X})$, es decir, encontrar la combinación de los parámetros de control tal que el efecto de los factores de ruido sobre el valor objetivo del sistema dinámico sea lo mas pequeño posible. Matemáticamente, dicho objetivo se expresa como:

$$\text{Min}_{x \in \Omega} : \sum_{i=1}^p k_i \frac{\sigma_{i\varepsilon}^2}{\beta_i^2} + \sum_{i=1}^p \sum_{i < j}^p \frac{k_{ij}}{\beta_i \beta_j} \left(\sigma_{Y_i Y_j} - \sigma_{Y_i^* Y_j^*} \right)$$

donde Ω es la región experimental del conjunto del factor control \mathbf{X}

Como hemos venido mencionando a lo largo del presente trabajo, Taguchi emplea las relaciones señal-ruido como medida de rendimiento del sistema, un mayor S/N significa un menor desvío de las respuestas con respecto a sus valores objetivos. También se ha dado un vistazo a los tipos de S/N según el tipo de variable, sin embargo un gran número de diferentes S/N han sido desarrollados para distintos problemas de entre los cuales los más importantes, según Jeyapaul y otros (2006), y de interés para nuestros objetivos son:

- Dinámico mayor es mejor (DLTB por sus siglas en *dynamic larger-the-best*)
- Dinámico nominal es mejor (DNTB por sus siglas en *dynamic nominal-the-best*)
- Dinámico menor es mejor (DSTB por sus siglas en *dynamic small-the-best*)

Definidos análogamente al tipo no dinámicos pero acorde con la pendiente deseada, es decir, $\beta = 0$, $0 < \beta < 1$ y $\beta = \infty$ respectivamente (Chang, 2008). Lo anterior se detalla en la siguiente sección.

2.1.4. Metodología Taguchi en sistemas no dinámicos

Uno de los aspectos más importantes de las relaciones dinámicas definidas por Taguchi, es considerar relaciones lineales y, como veremos más adelante, también es su punto más débil ya que en sistemas de producción las relaciones lineales son escasas (Zang y Mottershead, 2005). Para Taguchi, mejorar la linealidad hace que la salida del sistema (valor objetivo del producto) pueda ajustarse fácilmente, ya que la relación con la entrada es proporcional después de mejorar la S/N. Entonces, cuando puede ajustarse la salida a un determinado rango no es necesario realizar estudios redundantes sobre cualquier producto cuya salida esté dentro de dicho rango; ahorrando así tiempo y costes en el desarrollo. En esta gran ventaja radica el poder de la utilización de las relaciones S/N dinámicas (Wu y Wu, 1996), sin embargo en casos de múltiples respuestas un sistema dinámico puede ser considerado como un sistema no dinámico con múltiples respuestas donde varían sus valores señal (Joseph y Wu, 2002). Por todo lo anterior y a efectos del presente trabajo, a continuación se definen las relaciones S/N no dinámicas, conocidas también como señales S/N de tipo estático.

La filosofía de éste tipo de relaciones radica en que si se reduce la variabilidad y luego se ajusta la media al valor objetivo, no habrá otra circunstancia que hagan mover o ajustar la salida. Entonces, el enfoque no dinámico se utiliza para ajustar determinado producto al valor objetivo predeterminado, generalmente, por los factores señal. Hay 3 tipos de aplicaciones para las respuestas de salida no dinámicas, las cuales se mencionan a continuación.

Aproximación a un valor nominal: Generalmente reseñada en la literatura como *Nominal es mejor* (NTB). El objetivo es ajustar la media a un punto o valor nominal, por tanto se definen límites inferior y superior de especificación a ambos lados del objetivo. Entonces, para realizar el ajuste se considera un factor o parámetro que elimine el sesgo de la variable respuesta. Los parámetros de ajuste son factores controlables sobre las cuales se puede actuar, por tanto, por medio de ellos se puede reducir el sesgo. La pérdida media (2.1) viene dada entonces por $k\sigma^2$ y el objetivo será minimizar o reducir la variabilidad. Para este objetivo Taguchi propone la señal $S/N = -10 \log_{10}(S)$, donde S representa la desviación típica muestral calculada a partir de las n réplicas obtenidas para una combinación determinada de los niveles de los factores. Acotamos que el objetivo de minimizar S equivale a maximizar $S/N = -10 \log_{10}(S)$ y que en éste caso los factores control no afectan la varianza del sistema.

En casos contrarios, donde los parámetros de control modifican la media μ junto con la varianza σ^2 tal que el coeficiente de variación CV permanece constante, la pérdida media (2.1) se puede escribir como

$$E(L) = k[(CV)^2\mu^2 + (\mu - T)^2] \quad (2.7)$$

Se seleccionan entonces los niveles de los parámetros de ajuste que permiten aproximar o hacer coincidir μ con el valor nominal objetivo y entonces (2.7) se redefine como

$$E(L) = k[(CV)^2T^2$$

y a partir de tal expresión se estudian otros factores o parámetros de control que minimicen CV . A tales efectos, Taguchi define la señal

ruido

$$S/N = -10 \log_{10} \frac{\bar{y}^2}{S^2}$$

donde \bar{y} y S^2 denotan respectivamente la media y la varianza muestral de la variable respuesta para una combinación de los niveles de los parámetros en consideración.

Aproximación al valor más pequeño: Generalmente reseñada en la literatura como *Menor es mejor* (STB), aparecen cuando se desea minimizar la respuesta y el valor objetivo es cero, por tanto, la pérdida de calidad disminuye cuando la característica se acerca a cero. Entonces la pérdida media (2.1) queda como $E(L) = E(X^2)$, donde X es la característica de calidad bajo estudio y por tanto se usa la razón S/N como sigue

$$S/N = -10 \log_{10} \left(\frac{1}{n} \sum_{i=1}^n \bar{Y}^2 \right)$$

Aproximación al valor más grande: Reseñada en la literatura como *Mayor es mejor* (LTB). Aparecen cuando se desea maximizar el resultado, y el valor objetivo es infinito, por tanto, la pérdida de calidad disminuye cuando la característica crece. En este caso se considera la aproximación anterior en términos de la variable $\frac{1}{X}$ puesto que maximizar X equivale a minimizar $\frac{1}{X}$ y, por tanto, se define la razón S/N como

$$S/N = -10 \log_{10} \left(\frac{1}{n} \sum_{i=1}^n \frac{1}{\bar{Y}^2} \right)$$

2.2. Evolución de la metodología Taguchi

Las ideas de Taguchi constituyen grandes contribuciones a la ingeniería de calidad, sin embargo, desde sus inicios han dado lugar a grandes discusiones para su aplicación. Así por ejemplo, Box y Meyer (1986) muestran que en algunos casos es posible identificar factores que afecten tanto la varianza como la media empleando diseños factoriales fraccionados en lugar de los arreglos ortogonales sugeridos por Taguchi, poco después Ryan (1989) publica en su obra una discusión detallada de las limitaciones de los procedimientos empleados y recomienda el uso de algún tipo de diseño discriminante para eliminar factores no significantes e incorporar procedimientos de programación no lineal para la fase de optimización.

Tales discusiones han permanecido con el paso de los años, pero entre las más recientes podemos citar a Maghsoodloo y otros (2004), quienes en su obra muestran las fuerzas y limitaciones del enfoque de Taguchi en procesos de ingeniería, centrandose en que sus técnicas no son fáciles de aplicar en la vida real y considerando que las relaciones S/N carecen de rigurosidad estadística para identificar el mejor nivel del factor que minimice las pérdidas de calidad. Por su parte, Tsui (1996) y Montgomery (2001) afirman que el dividir en dos arreglos ortogonales trae como consecuencia el aumento en el número de repeticiones de los experimentos, lo que se traduce en costes innecesarios. Otros trabajos como los de Miller y Wu (1996); Su y Chang (2000); Zang y Mottershead (2005), entre otros, expresan que en la práctica el enfoque de Taguchi se limita a elegir el mejor nivel del factor entre un grupo especificado previamente, así como su limitación a parámetros de tipo discreto y el estudio de experimentos para disminuir el rango de los niveles de los factores de control, lo que trae un excesivo número de ensayos y falta de rigor en las conclusiones.

Diversas alternativas han surgido con el fin de mejorar el diseño de parámetros de Taguchi. Así por ejemplo, en el caso de una sola característica de calidad Chiu y otros (1997); Tay y Butler (1997) y Rowlands (1996)(citado por Chang, 2006) combinaron las ideas de Taguchi con el empleo de ANN pero dicha combinación resulta ineficiente en el momento de obtener la combinación óptima de parámetros (Su y Chang, 2000). Su y Chang (2000) proponen un enfoque que combina las ANN con SA y más recientemente Chang (2005) propone nuevamente combinar las ANN pero esta vez con GA.

Todos los aportes anteriores se limitan a estudios univariantes, pero actualmente en la optimización de procesos o productos es común considerar más de una característica de calidad, cada una reflejada por una variable respuesta. La metodología Taguchi sólo se limita a problemas de tipo univariantes (Zang y Mottershead, 2005), por tal razón podemos decir que éste punto también ha evolucionado y en la literatura encontramos propuestas para este tipo de problemas. A continuación se describen brevemente las más importantes.

Derringer y Suich (1980) y posteriormente Wu y Yeh (2005) proponen transformar las múltiples respuestas dentro de una función deseable (*desirability function*, introducidas por Harrington en 1965), la cual puede posteriormente ser optimizada mediante técnicas univariantes bajo ciertas consideraciones. Khuri y Conlom (1981) proponen minimizar una medida de distancia generalizada, definiéndola como la distancia ponderada de cada respuesta a su valor óptimo individual y los pesos son determinados por las varianzas y covarianzas de las respuestas. Logothetis y Haigh (1988) emplean técnicas de regresión múltiple y programación lineal, sin

embargo si los valores de prueba t de los parámetros de regresión resultan no significativos o el coeficiente de determinación R^2 resultase pequeño, el método sería limitado en su aplicación, además Tong y otros (1997) agregan que computacionalmente esta propuesta es compleja de aplicar por las implicaciones de programación que encierra y proponen una medida S/N donde fusiona las características de calidad en una sola respuesta, observando dificultad en determinar los pesos de cada respuesta en dicho ratio.

Mas tarde Koksoy (2008) emplea el criterio del MSE (cuadrado medio del error) en cada respuesta al considerarlas como independientes y no correlacionadas, lo que en términos reales resulta poco probable. Castillo y otros (1996) proponen una modificación a la función de deseabilidad de Harrington para optimizar múltiples respuestas, limitado a casos no diferenciables. Su y Tong (1997) proponen un método basado en el análisis de componentes principales (PCA), donde el conjunto de respuestas originales puede ser transformado en conjuntos no correlacionados y posteriormente ser optimizados; por tanto el conflicto para determinar el conjunto de combinaciones óptimas de los parámetros puede ser reducida. Antony (2001) adapta la función de pérdida de Taguchi al caso de múltiples respuestas mediante la asignación de pesos a cada característica de calidad. Kim y Lin (2000) presentan una modificación de la función exponencial deseable en el modelado de múltiples respuestas y posteriormente Hsu y otros (2004) emplean esas funciones exponenciales en combinación con las ANN para la optimización de un caso particular de múltiples respuestas. Surajit y Susanta (2010) proponen ajustar modelos de regresión múltiple a cada característica de calidad por separado para obtener la estimación de media y varianza que, posteriormente, permitirá maximizar la relación S/N ponderada; combinando así las ideas de Taguchi con técnicas de regresión

múltiple. Tong y Wand (citado por Wang y Tong (2005)) introducen la teoría del sistema Grey en combinación con la función de pérdida para la optimización de problemas dinámicos con múltiples respuestas; posteriormente Wang y Tong (2005) adoptan esa propuesta con una ligera modificación, agregando al sistema Grey la técnica TOPSIS desarrollada por Hwang y Yoon en 1981.

Todas las propuestas anteriores demostraron grandes habilidades en sus respectivas ilustraciones y constituyen grandes aportes en la mejora del diseño de parámetros. Sin embargo, desde el punto de vista práctico, éstos enfoques resultan un tanto complicados y laboriosos de aplicar (Chang, 2006) y además solo pueden obtener la mejor solución entre un grupo especificado de niveles de los factores control, es decir, son incapaces de lograr una combinación óptima si los factores de control poseen valores continuos.

La literatura ha ilustrado la potencia de las técnicas metaheurísticas y de minería de datos para resolver problemas que involucren factores de tipo continuo mediante los trabajos de Hou y otros (2006); Hsu y otros (2006) y los trabajos de Huang y Hung (2006); Su y otros (2005). Concretamente, para la mejora del diseño de parámetros Su y Chang (2000) proponen la combinación de ANN con SA en dos fases para el caso de una característica de calidad. En una primera fase emplean las ANN bajo el enfoque supervisado en virtud de no poseer un modelo a priori para el establecimiento de la función objetivo. La segunda fase se concentra en la optimización de los niveles de los factores de control mediante SA. Posteriormente, Chang (2005) modifica ligeramente tal propuesta al sustituir a SA por GA, atribuyendo buenas propiedades a GA como método de optimización.

Chang (2008) extiende el trabajo de Su y Chang (2000) a múltiples respuestas en cuatro fases. En la primera fase emplea ANN adaptado al caso de múltiples características de calidad. La segunda fase evalúa cada respuesta (medida de rendimiento PMs) usando funciones de deseabilidad exponencial. En la tercera fase los valores PMs son integrados en una medida OPI, definida como la media geométrica de las funciones de deseabilidad. Finalmente, en la cuarta fase, se emplea SA para obtener la combinación óptima de los niveles de los factores de control. Ahora bien, las ANN son técnicas potentes en el reconocimiento de patrones input/outputs, pero resultan costosas desde el punto de vista computacional e inestables en algunos casos. Por otra parte, existen algoritmos de optimización más eficientes que SA como lo expresan Chang y Chen (2011), quienes a mediados del año pasado modifican las propuestas anteriores en la última fase recomendando los algoritmos genéticos como método de optimización.

Finalizamos el presente capítulo acotando que consideramos que la combinación de las técnicas de minería de datos y de optimización heurística pueden seguir dando muy buenos resultados si se combinan eficientemente y por tal razón enmarcamos la línea del presente trabajo en proporcionar una alternativa completa, robusta y eficiente para la mejora del diseño de parámetros siguiendo los objetivos planteados en la sección 1.2.

Parte II

Estrategias para la modelización de sistemas

Capítulo 3

Introducción

El presente capítulo tiene como objetivo introducir los fundamentos teóricos que dieron origen a los esquemas de modelización que serán presentados posteriormente. Vale la pena mencionar que la base fundamental de nuestros esquemas se concentra en las estrategias que implican árboles de clasificación y regresión (CART, por sus siglas en inglés) y Random Forest (RF) bajo el enfoque del *aprendizaje supervisado*, pero hemos sumado a nuestro fundamento teórico algunas técnicas de remuestreo ya que serán incorporados a nuestros esquemas con el fin de mejorar la variabilidad propia de los árboles de regresión.

Para el caso de múltiples respuestas, nuestro esquema de modelización se fundamenta nuevamente en CART y RF como una generalización del caso univariante, lo que hemos definido como *random forest multivariante*. Proponemos métricas de modelaje, mediante el estudio de distintas funciones, que concentran las múltiples características de calidad (respuestas) por cada combinación de parámetros y que, bajo una secuencia de árboles de regresión univariante, tendrán como producto final las estimaciones de tales

respuestas. Sin embargo, el lector puede preguntarse porqué no usamos los árboles de regresión multivariante, entonces con el fin de despejar dicha duda mencionamos brevemente las limitaciones que presentan los árboles de regresión multivariante en los objetivos del presente trabajo y que hacen que no sea oportuna su incorporación en nuestros esquemas de modelización.

Para finalizar, en el presente capítulo describimos brevemente las bases teóricas de las redes neuronales artificiales (ANN), ya que forman parte de nuestra propuesta metodológica con fines comparativos, tanto en el caso univariante como en el caso multivariante.

3.1. Aprendizaje supervisado

La filosofía del aprendizaje supervisado (AS) es simple y se concentra en la siguiente idea. Supongamos que se establece un modelo

$$Y = f(X) + \varepsilon \tag{3.1}$$

donde Y es el atributo, variables o vector de variables a predecir, conocido como *outputs*, X es el vector p -dimensional (x_1, x_2, \dots, x_p) de las características que se toman para la predicción (variables explicativas), comúnmente llamado *inputs* y ε es el error aleatorio con $E(\varepsilon) = 0$. Entonces, el conjunto de observaciones, inputs-outputs, son unidos en un conjunto de entrenamiento (training) $T = (X, Y)$ y son procesados en un sistema artificial, que normalmente lo constituyen programas computacionales conocidos como *algoritmos de aprendizaje* que producen outputs estimados, es decir $\hat{f}(X)$ en respuesta a los inputs. Cabe mencionar que el algoritmo de aprendizaje puede modificar la relación (3.1) en respuesta a la diferencia entre el output original y el output generado, es decir

$y_i - \hat{f}(x_i)$. Lo anterior es conocido como *aprendizaje por muestras* (Learning by sample). Cuando el proceso de aprendizaje culmina, se espera que tanto los outputs reales como los artificiales estén muy próximos, a fin de ser usados para fines predictivos.

Como veremos en nuestra propuesta de modelación, la elección del algoritmo de aprendizaje adecuado es fundamental en la filosofía del AS, ya que debe ser capaz de recoger toda la información del conjunto de datos, es decir, ser capaz de reconocer las relaciones estructurales complejas y de interacciones entre las observaciones que permitan establecer de forma eficiente el modelado a fines predictivos. Las ANN inspiradas en la estructura y funcionamiento del cerebro humano, y CART mediante su estructura de grafos acíclicos, son algoritmos de aprendizaje capaces de identificar intrínsecamente relaciones no lineales del conjunto de datos inicial. Por tal razón y por constituir nuestra base en la propuesta de modelización exponemos brevemente sus fundamentos teóricos

3.2. Árboles de clasificación y regresión

Supongamos una variable respuesta Y , categórica, con valores en un conjunto $\mathcal{C} = \{C_1, \dots, C_{j-1}\}$ de clases y \mathbb{X} el conjunto de todos los valores del vector X de variables predictoras. Entonces, el objetivo será construir una regla de clasificación que le asigne a cada observación $x \in \mathbb{X}$ una de las clases de \mathcal{C} .

La regla óptima de clasificación es la que maximiza la probabilidad a posteriori:

$$P(Y = C_i / X = x), (i = 0, 1, \dots, j - 1)$$

es decir, a cada observación $x \in \mathbb{X}$, le asigna la clase que da la mayor pro-

babilidad a posteriori. Dicha regla, fundamentada en la teoría de decisión, divide a \mathbb{X} en j regiones disjuntas R_0, \dots, R_{j-1} tales que $\mathcal{C} = \bigcup_{i=0}^{j-1} R_i$ y cada $x \in R_i$ se le asigna la clase C_i .

Cuando se asume la hipótesis de que la distribución del vector X en cada una de las clases es una normal multivariante con la misma matriz de covarianzas para todas las clases, las fronteras entre las regiones anteriores son lineales y el clasificador óptimo se denomina *lineal*. Ahora bien, las fronteras de dicho clasificador se estiman a partir de los datos por lo que es necesario contar con buenas estimaciones de cada uno de los parámetros del modelo y la validez de dichas estimaciones dependerá del cumplimiento de los supuestos iniciales de homocedasticidad y normalidad del vector X . Por tanto, verificar el cumplimiento de estos supuestos antes de establecer el clasificador lineal es fundamental pero no siempre factible. Entonces, surgen los árboles de clasificación como una alternativa no paramétrica al clasificador lineal.

En términos más generales, CART constituye una técnica no paramétrica basada en la generación de un modelo con estructura de grafos acíclicos (árbol) que permita explicar o predecir una determinada variable respuesta que puede ser tanto categórica como continua (Breiman y otros, 1984).

El algoritmo está basado en el particionamiento recursivo de X en regiones disjuntas, llamadas *nodos*, y en la asignación de una clase a cada una de las regiones que resultan del proceso de segmentación. Así, el *nodo raíz* que representa a toda la población es dividido en subgrupos determinados por la partición de una variable predictora elegida generándose nuevos nodos, *nodos intermedios*, que son divididos a su vez por otra variable predictora, calculándose nuevamente la media de la variable respuesta en el nodo.

El proceso recursivo se repite sucesivamente hasta que se cumpla alguna condición de parada, donde los nodos que no poseen divisiones se definen *nodos terminales*. La figura 3.1 ilustra el proceso anterior para X_1 y X_2 , donde la región $X_1 = t_1$ representa el nodo raíz el cual es dividido por t_1 en las regiones $X_1 \leq t_1$ y $X_1 > t_1$. La región $X_1 \leq t_1$ es dividida en $X_2 = t_2$ y $X_1 > t_1$ es dividida en $X_3 = t_3$. Finalmente la región $X_1 > t_3$ es dividida en $X_2 = t_4$. El resultado de éste proceso de partición consta de las regiones $(R_1, R_2, R_3, R_4, R_5)$ dado lugar al modelo de predicción de Y con constante c_m en la región R_m , esto es

$$\hat{f}(X) = \sum_{m=1}^5 c_m I\{(X_1, X_2) \in R_m\}$$

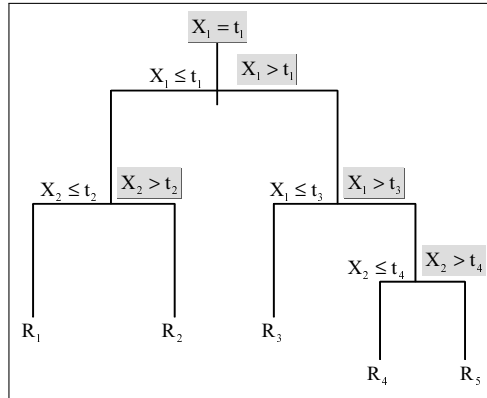


Figura 3.1: Árbol de regresión para $\mathbb{X} = \{X_1, X_2\}$

Vale la pena acotar que, en la figura 3.1, las expresiones sombreadas normalmente quedan implícitas en los gráficos. En esta oportunidad las hemos colocado sólo con fines ilustrativos.

En los casos de árboles de regresión, Y es cuantitativa y el objetivo es

construir una regla de predicción del comportamiento de la respuesta en base a la observación del vector X de las variables explicativas. La regla es asignar a cada observación $x \in X$ un valor de la respuesta c . Para describir, de forma general, la formación de los árboles de regresión, supongamos un conjunto de N observaciones (x_i, y_i) con $i = 1, \dots, N$ y $x_i = (x_{i1}, \dots, x_{ip})$ para una variable respuesta. Entonces el algoritmo comienza con todo el conjunto de datos, se considera una variable de división j en un punto s resultando dos semiplanos o *nodos descendientes*:

$$R_1(j, s) = \{X/X_j \leq s\} \quad R_2(j, s) = \{X/X_j > s\}$$

entonces se busca la variable de división j en el punto s que minimice el error cuadrático medio, es decir, que dé solución a la expresión

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right] \quad (3.2)$$

Para cualquier j y s en (3.2), la minimización interna se resuelve como sigue

$$\hat{c}_1 = \text{ave}(y_i/x_i \in R_1(j, s)) \quad \hat{c}_2 = \text{ave}(y_i/x_i \in R_2(j, s))$$

es decir, \hat{c}_1 y \hat{c}_2 representan la media de todas las respuestas y_i de x_i en las regiones $R_1(j, s)$ y $R_2(j, s)$, respectivamente. Éste proceso es realizado sobre el umbral de las N observaciones, determinando así el mejor par (j, s) . Entonces, bajo dicha división, nuevamente se particiona el conjunto de datos en dos regiones y se repite el proceso. Las divisiones se realizan de forma recursiva hasta obtener las M regiones establecidas o cumplir alguna condición de parada. Pero, ¿Cuántas divisiones realizar? ó ¿Cómo establecer una condición de parada?, pues es claro que un árbol con gran número de divisiones podría sobreajustar los datos y en caso contrario, es

decir, en un número muy escaso de divisiones se escaparía la importancia de la estructura.

En respuesta a dichas interrogantes, Breiman y otros (1984) recomiendan crear el árbol lo más grande posible (todas las divisiones posibles) hasta llegar a un mínimo tamaño de nodo, 2 observaciones por ejemplo, y luego podar dicho árbol estudiando lo que define como *criterio de coste-complejidad*, que penaliza estructuras complejas de árbol grandes y permite obtener una secuencia de subárboles provenientes del más grande (T_{max}), es decir $T_0 \subset \dots \subset T_{max}$. La *complejidad* se refiere al número de nodos terminales del árbol. A continuación se detalla dicho procedimiento bajo la óptica de Hastie y otros (2009).

En efecto, consideremos la indexación de los nodos terminales con m tal que con el nodo m se representa la región R_m , $|T|$ representa el número de nodos terminales de T y T es un subárbol cualquiera que resulta de la poda del árbol máximo, es decir $T \subset T_{max}$.

Sean:

$N_m = \#\{x_i \in R_m\}$, el número de casos que contiene el nodo,

$\hat{c}_m = \frac{1}{N_m} \left(\sum_{x_i \in R_m} y_i \right)$, la media de todas las respuestas de x_i en R_m , y

$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2$, la varianza muestral del nodo.

Entonces se define el criterio de coste-complejidad como:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T| \quad (3.3)$$

Donde el objetivo es determinar, para cada $\alpha > 0$, el subárbol $T_\alpha \subseteq T_{max}$ que minimice a $C_\alpha(T)$. Entonces el ajuste del parámetro α dependerá directamente de la relación entre el tamaño del árbol y el error relativo (bondad de ajuste a los datos), por tanto se deben emplear estimaciones fiables de dicho error. Vale la pena mencionar que, para el desarrollo de nuestros modelos de predicción, el error relativo será estimado mediante validación cruzada y denotaremos a α como *cp*.

Como se ha podido ver, los árboles de regresión proporcionan, además de un modelo de predicción, un patrón de comportamiento de las observaciones pertenecientes a cada una de las clases; lo cual es útil para interpretar la estructura que acompaña al conjunto de datos. No obstante, bajo el enfoque de AS la información dada por la estructura del árbol esta sometida a perturbaciones aleatorias que tienen su causa, muchas veces, en la determinación de los conjuntos de entrenamiento y prueba. Así pequeños cambios en dichos conjuntos pueden originar árboles muy distintos. Puesto que las técnicas de remuestreo constituyen una poderosa herramienta para paliar la variabilidad de las muestras, hemos decidido combinar técnicas como validación cruzada, Bootstrap y Bagging con los algoritmos CART. Por tal razón, a continuación se presentan brevemente sus aspectos más relevantes.

Validación cruzada (VC)

Este método, comúnmente conocido como *k-fold cross validation*, compara algoritmos mediante la división del conjunto de datos en *k* partes iguales.

En general, estos conjuntos son cruzados k veces, tal que, dentro de cada iteración una parte de las partes (*fold*) es usado como conjunto de evaluación mientras que el resto ($k-1$) es usado como entrenamiento; así, cada punto en el conjunto de datos es evaluado en el modelo.

El método es bastante simple y se ilustra mediante la figura 3.2 considerando $k = 5$, donde las partes más oscuras ($k - 1 = 4$) son usadas para el entrenamiento del modelo, mientras que la más clara ($k - 4 = 1$) es usada para la validación.

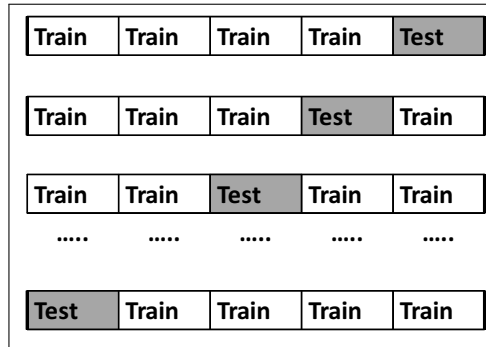


Figura 3.2: Esquema general para validación cruzada con $k=5$

Bajo el enfoque de AS, la validación cruzada es usada para evaluar y comparar uno o más algoritmos en cada iteración, esto es, en cada iteración uno o más algoritmos usa $k - 1$ partes para comparar uno o más modelos. Una vez establecido el modelo se realizan las predicciones con la parte de prueba y se comparan los rendimientos de dichos modelos mediante medidas y criterios previamente establecidos. Vale la pena acotar que el valor k es fijado por el usuario, pero autores como Breiman y Spector (1992) y Kohabi (1995) recomiendan tomar $k \in [5, 10]$ cuando el objetivo es minimizar la varianza del aprendizaje.

Bootstrap

Este método propuesto por Efron y Tibshirani (1996), en su enfoque no paramétrico constituye una poderosa herramienta para estimar el error estadístico y se fundamenta en el teorema de Glivenko-Cantelli, el cual establece una convergencia casi segura pero asintótica entre una distribución desconocida F (del parámetro de interés) y una empírica F_n , calculada a partir de una muestra cuando $n \rightarrow \infty$.

Para explicar en qué consiste el método supongamos un conjunto de entrenamiento $\mathbf{Z} = (z_1, \dots, z_N)$ tal que $z_i = (x_i, y_i)$, con distribución F desconocida. La idea básica es extraer muestras aleatorias con reemplazamiento de tamaño N , es decir cada muestra tendría el mismo tamaño de \mathbf{Z} . Dicho remuestreo se realiza el número de veces establecido previamente, b -veces por ejemplo, obteniéndose las llamadas *muestras bootstrap* que denotaremos como \mathbf{Z}^* y se ajusta el estadístico de interés, supongamos $\hat{\theta}$, a cada muestra bootstrap. De esta forma se tienen b -estimaciones del estadístico de interés que permitirán establecer su distribución empírica $F_b(\hat{\theta})$ para el estudio de las estimaciones correspondientes y el error estadístico. El procedimiento anterior se ilustra con la figura 3.3.

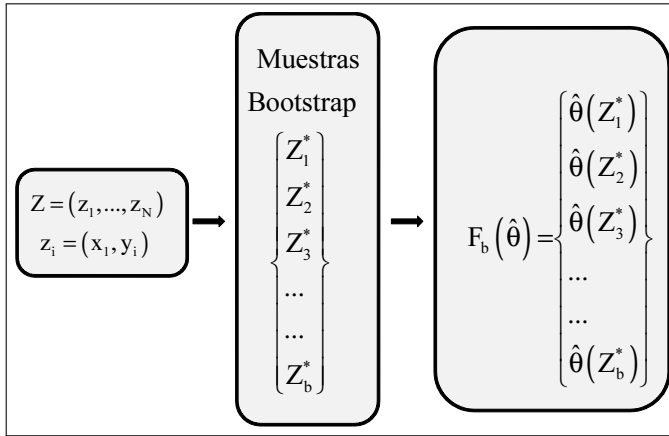


Figura 3.3: Algoritmo general Bootstrap

Bagging

Anteriormente presentamos a Bootstrap como método de remuestreo que permite estudiar el rendimiento del estadístico de interés o las predicciones bajo una distribución empírica. Ahora bien, dichas muestras son tomadas con reemplazamiento lo que puede enmascarar el error estadístico ya que muestras diferentes pueden poseer las mismas respuestas o la misma respuesta varias veces. Bagging, propuesta por Breiman (1996) se dota de la idea aportada en Bootstrap pero ahora las muestras son tomadas sin reemplazo y el estimador será el promedio del ajuste del estadístico de interés a cada muestra tomada, de esta forma se reduce la variabilidad que pueda existir entre dichas muestras (Sutton, 2005). Entonces, supongamos nuevamente se tiene un conjunto de entrenamiento $\mathbf{Z} = (z_1, \dots, z_N)$ tal que $z_i = (x_i, y_i)$ pero ahora se extraen las b -muestras sin reemplazo y de tamaño n^* tal que $n^* < N$. Se ajusta a cada muestra el estadístico de interés $\hat{\theta}$ y la estimación final estará determinada por el promedio de dichos ajustes, lo que matemáticamente expresamos como sigue y se ilustra con la figura 3.4.

$$\hat{\theta}_{bagg} = \frac{1}{b} \sum_{i=1}^b \hat{\theta}(Z_i^*)$$

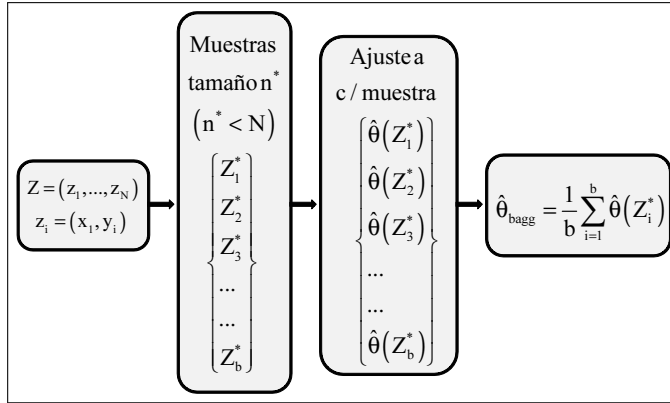


Figura 3.4: Algoritmo general Bagging

3.2.1. Random Forest

La metodología de Random Forest (RF) se basa en la construcción de árboles de predicción mediante el empleo Bootstrap y Bagging, lo que hace el proceso estable (Breiman, 2001). Básicamente la idea se concentra en los siguientes puntos, los cuales se comentan a detalle posteriormente:

- No se genera un único árbol sino un gran número de árboles, todos ellos sin poda.
- Los árboles se construyen a partir de muestras bootstrap con reposición, con la finalidad de corregir el error de predicción a consecuencia de la selección específica de una muestra, así como para disponer por cada árbol de una muestra independiente (*out-of-bag*) para la estimación del error de clasificación, puesto que

aproximadamente un tercio de la muestra original queda excluida de cada muestra generada por bootstrap.

- Para cada división de un nodo, no se selecciona la mejor variable de entre todas como en CART, sino que se selecciona al azar un conjunto de variables de un tamaño previamente establecido y se restringe la selección de la variable de división a dicho conjunto. De esta forma se incluye una mayor variabilidad de árboles y se reduce la dependencia del resultado con las divisiones precedentes.
- A diferencia de CART, no proporciona una representación gráfica de las interrelaciones entre las variables, sino que establece un *ranking* de la importancia de las variables en la predicción de la variable respuesta.

El proceso *out-of-bag* (OOB) consiste en usar el conjunto de entrenamiento \mathbf{T} para crear k muestras bootstrap de entrenamiento \mathbf{T}_k , se construyen los árboles $h(\mathbf{x}, \mathbf{T}_k)$ y el promedio de ellos será el predictor bagged. Posteriormente, para cada (y, \mathbf{x}) de \mathbf{T} se construyen los árboles en cada \mathbf{T}_k que no contienen a (y, \mathbf{x}) , es decir, las muestras que quedaron fuera de las muestras bootstrap, siendo éstos los clasificadores OOB que permitirán estimar el error de clasificación sobre el conjunto \mathbf{T} , semejando esto al conjunto de prueba de los métodos presentados anteriormente. Las muestras OOB también son usadas en RF para calcular la fuerza de predicción de cada una de las variables usadas, conociéndose esto como *importancia de las variables*, que está condicionada a su interacción con el resto de las variables. RF calcula dos medidas de importancia diferentes; MDA (*Mean decrease accuracy*) y MDG (*Mean Decrease Gini*).

MDA se basa en la contribución de la variable al error de predicción, es decir, el porcentaje de mal clasificados. El error de clasificación de

cada árbol se calcula a partir de la parte de la muestra que ha quedado excluida de la submuestra usada en la construcción del árbol generada por el remuestreo. Entonces, para calcular la importancia de cada una de las variables que aparecen en un árbol se permutan aleatoriamente los valores de esa variable, dejando intactos el resto de variables, y se vuelven a clasificar los mismos individuos según el mismo árbol pero ahora con la variable permutada. La importancia en ese árbol se calcula como el aumento en el error de predicción resultante y, finalmente, la medida MDA será la media de estos incrementos en todos los árboles en los que participa la variable permutada.

La segunda medida de importancia, MDG, se calcula a partir del índice *Gini*. Éste es el criterio que se utiliza para seleccionar la variable de cada partición en la construcción de los árboles para el caso de variables respuestas categóricas y que comporta una disminución de esta medida. Entonces la importancia de una variable en un árbol se mide como la suma de los decrementos atribuidos a esa variable y finalmente la medida MDG será la media en todos los árboles.

Por otro lado, autores como Segal (2003); Siroky (2005); Hastie y otros (2009) atribuyen como una gran ventaja de RF la reducción de la dependencia entre árboles en la determinación de los nodos mediante la elección aleatoria de conjuntos de predictores en cada árbol. En torno a esto, Breiman (2001) sugiere determinar previamente el número de variables a elegir en cada nodo (m_{try}). Específicamente, para el caso de árboles de regresión recomienda $m_{try} = \frac{p}{3}$, donde p es el número de predictores en la base de datos. Vale la pena acotar que, como veremos en nuestra propuesta de modelización, dicho parámetro puede ser optimizado mediante validación cruzada u OOB.

Finalmente, a efectos del presente trabajo se ilustra el algoritmo RF mediante la figura 3.5 para $B = 1000$ y se detallan sus pasos a continuación.

Algoritmo general RF

- Se toman B muestras bootstrap de tamaño N del conjunto de entrenamiento.
- Se crean T_b , ($b = 1, \dots, B$) árboles con las muestras hasta que se obtiene el tamaño mínimo en el nodo terminal. Esto se logra de forma recursiva mediante los siguientes pasos:
 1. Seleccionar aleatoriamente m_{try} variables del conjunto total de P variables.
 2. Seleccionar la óptima variable de división entre las p variables.
 3. Dividir el nodo en dos nodos hijos.
- El conjunto de salida es el ensamble (promedio) de los $\{T_b\}_1^B$ árboles, es decir

$$\hat{f}_{RF}^B(x) = \frac{1}{B} \sum_{b=1}^B T_b(x)$$

- La estimación de la tasa de error o error de clasificación se obtiene mediante el conjunto OOB.

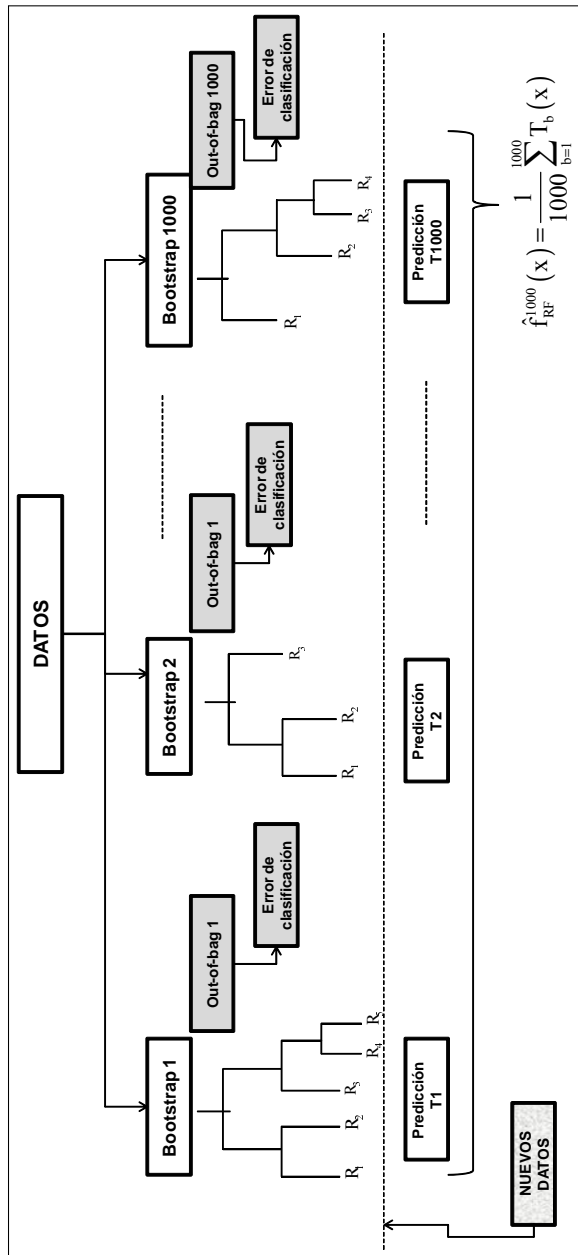


Figura 3.5: Algoritmo Random Forest para $b = 1000$

Hasta ahora hemos presentado las estrategias que fundamentan teóricamente nuestra propuesta de modelización desde el punto de vista univariante. Como hemos mencionado en los objetivos del presente trabajo, nuestra propuesta metodológica se extiende al caso multivariante por lo que obviamente el lector puede suponer que nuestro fundamento se enmarcará en árboles de regresión multivariante. De hecho ésa fue nuestra idea inicial, pero tales árboles presentan fuertes limitaciones en cuanto a las características que deseamos en nuestra función objetivo. Para aclarar mejor este punto, a continuación damos una breve descripción de dicha metodología.

3.2.2. Árboles de regresión multivariante

Los árboles de regresión multivariante (MRT) constituyen una extensión directa de CART, donde el resultado está dado por árboles cuyos nodos terminales están compuestos por subgrupos que minimizan la suma de cuadrados intra-grupo pero donde la siguiente partición está definida por el valor del umbral de una de las variables explicativas (De'ath, 2002). El procedimiento general se resume a continuación.

- Para cada variable explicativa se producen todas las posibles particiones de los sitios en dos grupos, en casos de variables cuantitativas ésto es realizado por sorteo de acuerdo con el orden de los valores en la variable, y en casos de variables categóricas se distribuyen los objetos en dos grupos tal que se proyecten todas las posibles combinaciones de los niveles. En todos los casos se calcula por cada partición la suma de cuadrados de las desviaciones a la respuesta media del grupo en el vector de respuestas (*suma de cuadrados intra-grupo*). Se tomará aquella división que minimice la suma de cuadrado intra-grupo, produciéndose así los primeros *nodos*

descendientes.

- El proceso es repetido dentro de cada nodo descendiente formado en el paso anterior, resultando la mejor partición aquella que nuevamente minimize la suma de cuadrados intra-grupo en el umbral de la variable del nodo. Así el proceso es repetido reiteradas veces hasta llegar a un tamaño preestablecido o alguna condición de parada.

Ahora bien, análogamente a CART vemos como cada nodo descendiente dependerá directamente del umbral marcado por el nodo anterior, lo que lo hace sensible a las variaciones que puedan crearse en la determinación de los conjuntos de entrenamiento y prueba. Sumado a ello tenemos que un punto determinante en la formación de los nodos corresponde a la media de las respuestas dentro de cada grupo, por tanto escalas muy diferentes entre las variables respuestas también afectará notoriamente a la estructura del árbol.

Ante tal limitante, en un reciente estudio, Dine y otros (2009) proponen un método para la construcción de MRT creando *variables sustitutas* producto del agrupamiento de variables respuesta del mismo tipo, resaltando los beneficios del método sólo a fines descriptivos y dejando abierto el estudio para casos predictivos por la herencia de variabilidad que conlleva la construcción de un sólo árbol. Tal variabilidad nos hace pensar que establecer un solo árbol para modelar casos multivariantes no resultaría estable estadísticamente y por tanto no se cumpliría uno de nuestros objetivos iniciales. Considerando que la esencia de RF es el manejo de colecciones de árboles, hemos decidido ajustar su algoritmo a casos de múltiples respuestas, lo cual se tratará en detalle en el capítulo correspondiente.

3.3. Redes Neuronales Artificiales

Las ANN constituyen un sistema compuesto de procesadores simples, conectados en paralelo, cuya función es determinada por la estructura de la red. Su funcionamiento general se ilustra con la figura 3.6, donde se presenta una neurona con 4 entradas (x_1, x_2, x_3, x_4) . Cada x_i representa la señal que llega de una o más neuronas y la salida puede ser la señal emitida por la neurona a otra o el vector de respuestas, Y . Las fuerzas entre las conexiones de las neuronas, conocida como fuerza sináptica, es representada por los *pesos* w_i ($i = 1, \dots, 4$). Éstas conexiones son dirigidas, es decir, la información se propaga en un sólo sentido y los pesos son números reales. El proceso interno de la neurona está a cargo de una función de propagación (también conocida como función de excitación o de transferencia), que por lo general consiste en el sumatorio de cada entrada multiplicada por el peso de su interconexión (valor neto). Si el peso es positivo, la conexión se denomina excitatoria; si es negativo, se denomina inhibitoria. Finalmente, el valor devuelto por la neurona es procesado por una función de transferencia, la cual se usa para acotar la salida de la neurona y generalmente viene dada por la interpretación que queramos darle a dichas salidas. Algunas de las más utilizadas son la función sigmoideal (para obtener valores en el intervalo $[0, 1]$) (Hastie y otros, 2009), cuya expresión matemática viene dada por (3.4).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.4)$$

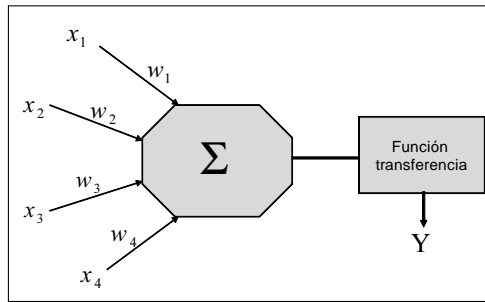


Figura 3.6: Funcionamiento de una neurona en ANN

Los pesos pueden ser de tipo adaptable o fijo. Los tipo adaptable emplean leyes de aprendizaje para ajustar el valor de la fuerza de interconexión con otras neuronas, mientras que los pesos de tipo fijo emplean leyes previamente establecidas. A efectos del presente trabajo, se emplearan pesos de tipo adaptable. Las neuronas pueden agruparse en unidades estructurales llamadas *capas*, las cuales determinan la estructura o arquitectura de la red en mono o multi capa. En el caso más estándar, como lo muestra la figura 3.7 y considerando igual número de neuronas por capa, las neuronas de la red se dividen en varias capas, q representa el número de neuronas del vector de entrada, m el número de neuronas en la capa oculta y l el número de neuronas en la capa de salida; w_i representan los pesos de la i -ésima capa de entrada a la i -ésima capa oculta y w_j el de la i -ésima capa oculta a la i -ésima capa de salida. Acotamos que el usuario debe establecer previamente el número de neuronas de la capa de entrada y salida, el cual dependerá del número de componentes del vector de entradas y salidas correspondientemente, además determinar la cantidad de capas ocultas y el número de neuronas de cada capa, las cuales pueden diferir de capa en capa. Otra característica importante es que las salidas están restringidas al intervalo $(0, 1)$, por lo que es necesario que el conjunto de entrada sea normalizado en dicho intervalo por las funciones

de transferencia, las cuales pueden diferir de capa a capa. Software como Qnet2000 (2000) normalizan automáticamente la base de datos, dejando los detalles ocultos al usuario. Finalmente y bajo el enfoque AS, las ANN permiten fijar el número de iteraciones en la fase de entrenamiento, las cuales se definen como *epochs*.

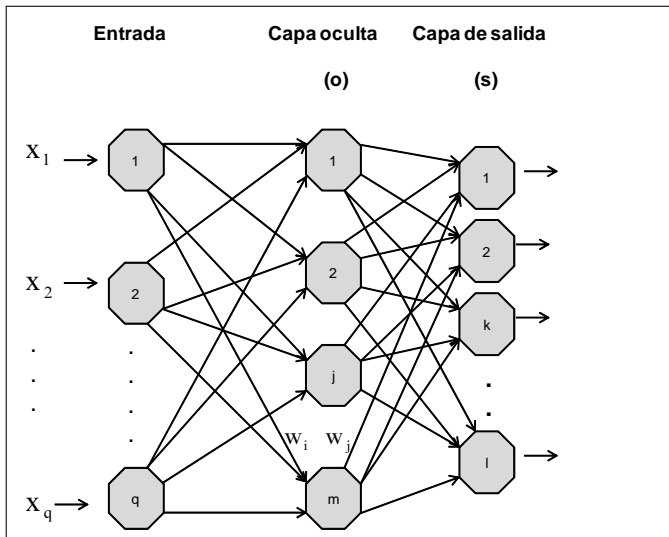


Figura 3.7: Estructura estándar de una red neuronal

Existen diferentes tipos de ANN, dependiendo de la estructura que posean. De entre las más nombradas en la literatura se encuentran el Perceptrón, Perceptrón multicapa, Backpropagation (Propagación inversa), (Hastie y otros, 2009). Considerando los objetivos del presente estudio y a fin de tener una idea matemática del caso más simple, se describe brevemente el algoritmo de ANN en propagación inversa considerando una capa de entrada, una capa oculta y una capa de salida; tal como se ilustró en la figura 3.7. En efecto, este algoritmo para redes multicapa, se caracteriza por la actualización de los pesos con base al error cuadrático medio. Trabaja bajo AS por lo que es necesario definir un conjunto de entre-

namiento que describa la entrada y salida esperada de la red de la forma $\{(p_1, t_1), \dots, (p_q, t_q)\}$, donde el primer elemento de cada par representa una entrada de la red y el segundo elemento es la correspondiente salida para cada patrón q -ésimo. Así, el algoritmo debe ajustar los parámetros de la red para minimizar el error cuadrático medio.

Supongamos el vector de entrada con q componentes $P = \{p_1, \dots, p_q\}$. Para iniciar el entrenamiento, dicho vector se propaga a través de las conexiones existentes, produciendo una entrada neta n en cada una de las neuronas de la siguiente capa; por ejemplo, la entrada neta a la neurona j de la siguiente capa debido a la presencia del vector de entrenamiento viene dada por

$$n_j^0 = \sum_{i=1}^q w_{ij}^0 p_i + b_j^0 \quad (3.5)$$

donde w_{ij}^0 es el peso que une a la componente i de la entrada con la neurona j de la capa oculta, p_i es la componente i del vector P con q componentes y b_j^0 es la ganancia de la neurona j en la capa oculta. El superíndice $\{0\}$ representa la capa a la que pertenece cada parámetro, que en este caso representa la capa oculta. La salida de cada una de las neuronas de la capa oculta se denotará como

$$a_j^0 = f^0(n_j^0) = f^0\left(\sum_{i=1}^q w_{ij}^0 p_i + b_j^0\right) \quad (3.6)$$

donde f^0 representa la función de transferencia de las neuronas de la capa oculta. Entonces, las salidas a_j^0 de las neuronas de la capa oculta (de m componentes) son las entradas a los pesos de conexión de la capa de salida, entonces (3.5) queda como

$$n_k^s = \sum_{j=1}^m w_{kj}^s a_j^0 + b_k^s \quad (3.7)$$

donde ahora, w_{ij}^s es el peso que une a la neurona j de la capa oculta con la neurona k de la capa de salida, formada por las s neuronas; b_j^s es la ganancia de la neurona k de la capa de salida y n_k^s será la entrada neta a la neurona k de la capa de salida. Finalmente la ANN produce la salida a_k^s como en (3.6) tal que $a_k^s = f^s(n_k^s)$, entonces de (3.7) la salida de la red en función de la entrada neta y los pesos de conexión con la última capa oculta viene dada por la expresión

$$a_k^s = f^s \left(\sum_{j=1}^m w_{kj}^s a_j^0 + b_k^s \right)$$

donde f^s sera la función de transferencia de las neuronas de la capa de salida. La salida de la red de cada neurona a_k^s se compara con la salida deseada t_k para calcular el error de cada unidad, es decir $\delta_k = \frac{(t_k - a_k^s)}{2}$, entonces el error cuadrático medio para cada patrón propagado es

$$ep^2 = \frac{1}{2} \sum_{k=1}^l (t_k - a_k^s)^2 = \frac{1}{2} \sum_{k=1}^l \delta^2 \quad (3.8)$$

El proceso anterior se repite para el número total de patrones de entrenamiento r . Para un proceso de aprendizaje exitoso el objetivo del algoritmo es actualizar todos los pesos y ganancias de la red minimizando el error cuadrático medio total, entonces de (3.8) se deduce el error cuadrático medio, se denota como sigue y representa el error total del aprendizaje en una iteración.

$$e^2 = \sum_{p=1}^r ep^2 = \sum_{p=1}^r \left(\frac{1}{2} \sum_{k=1}^l \delta^2 \right)$$

El error que genera una red neuronal en función de sus pesos genera un espacio de n dimensiones, donde n es el número de pesos de conexión de la red. Al evaluar el error en un punto de ésta superficie se obtendrá la dirección en la cual la función del error tendrá un mayor crecimiento. Como el objetivo del proceso de aprendizaje es minimizar el error, debe tomarse la dirección negativa del gradiente para obtener el mayor decremento del error y de esta forma su minimización, condición requerida para la actualización de pesos en este algoritmo. El gradiente negativo de ep^2 se calcula como la derivada del error respecto a todos los pesos de la red, esto es:

$$w_{k+1} = w_k - \delta \nabla ep^2 \Rightarrow -\frac{\partial ep^2}{\partial w_{kj}^s} \left(\frac{1}{2} \sum_{k=1}^l (t_k - a_k^s)^2 \right) = (t_k - a_k^s) \frac{\partial a_k^s}{\partial w_{kj}^s}$$

donde $-\frac{\partial ep^2}{\partial w_{kj}^s}$ es el componente del gradiente $-\nabla ep^2$ respecto al peso de la conexión de la neurona de la capa de salida y la neurona j de la capa oculta w_{kj}^s y $\frac{\partial a_k^s}{\partial w_{kj}^s}$ es la derivada de la salida de la neurona k de la capa de salida respecto al peso w_{kj}^s y para su cálculo se emplea la regla de la cadena, resultando finalmente la expresión:

$$-\frac{\partial ep^2}{\partial w_{kj}^s} = (t_k - a_k^s) f'^s(n_k^s) a_j^0$$

Hemos de mencionar que la expresión anterior restringe las funciones de transferencia a funciones continuas para que su derivada exista en todo el intervalo. Así mismo, de aquí se desprenden los términos de error para las neuronas de la capa de salida $\delta_k^s = (t_k - a_k^s) f'^s(n_k^s)$.

El algoritmo se denomina *propagación inversa* debido a que el error se propaga de manera inversa al funcionamiento normal de la red. De esta forma, el algoritmo encuentra el error en el proceso de aprendizaje desde

las capas más internas hasta llegar a la entrada; con base en el cálculo de este error se actualizan los pesos y ganancias de la capa. Así pues, se procede a calcular el error en la capa oculta mediante la expresión:

$$-\frac{\partial ep^2}{\partial w_{ji}^0} = -\frac{\partial}{\partial w_{ji}^0} \left(\frac{1}{2} \sum_{k=1}^l (t_k - a_k^s)^2 \right) = \sum_{k=1}^l (t_k - a_k^s) \frac{\partial a_k^s}{\partial w_{ji}^0}$$

Como la salida de la red no es una función explícita de los pesos de la conexión entre la capa de entrada y la capa oculta, entonces el último término de la expresión anterior debe calcularse empleando la regla de la cadena reiteradas veces. Entonces la expresión del gradiente del error en la capa oculta es

$$-\frac{\partial ep^2}{\partial w_{ji}^0} = -\sum_{k=1}^l (t_k - a_k^s)^2 f'^s(n_k^s) w_{kj}^s f'^0(n_j^0) p_i$$

Los términos del error para cada neurona de la capa oculta vienen dadas por $\delta_j^0 = f'^0(n_j^0) \sum_{k=1}^l \delta_k^s w_{kj}^s$, donde δ_j^0 se le define como la sensibilidad de la capa. Para finalizar el proceso, después de determinar el gradiente del error se procede a actualizar todos los pesos en todas las capas, empezando por la salida y luego los de la capa oculta. Las expresiones correspondientes son:

$$\begin{aligned} w_{kj}^s(t+1) &= w_{kj}^s(t) - 2\alpha\delta_k^s & b_k(t+1) &= b_k(t) - 2\alpha\delta_k^s \\ w_{kj}^0(t+1) &= w_{kj}^0(t) - 2\alpha\delta_j^0 & b_j(t+1) &= b_j(t) - 2\alpha\delta_j^0 \end{aligned}$$

donde α representa la tasa de aprendizaje y varía en el intervalo $(0, 1)$, dependiendo de las características del problema en particular.

Modelización para casos univariantes

4.1. Esquema general de modelización

Para el problema del diseño de parámetros en el presente capítulo, suponemos que la respuesta y_{ijkl} esta determinada por

$$y_{ijkl} = f_i(\mathbf{X}_k, M_j, Z_l) + \varepsilon_{ijkl} \quad (4.1)$$

donde $f_i(\mathbf{X}_k, M_j, Z_l)$ representa la función de la $ijkl$ -ésima respuesta y la correspondiente k -ésima combinación del vector de factores control, el j -ésimo nivel del factor señal y el l -ésimo nivel del factor ruido; mientras que ε_{ijkl} representa el error aleatorio. El objetivo, en esta fase, corresponde a la modelización de la respuesta (función objetivo) minimizando su sensibilidad a los factores de ruido.

Los algoritmos propuestos en la presente sección, se ajustan a estrategias

basadas en el uso de CART y RF. No obstante, para poder realizar las comparaciones adecuadas hemos realizado también pruebas numéricas con el algoritmo propuesto por Su y Chang (2000), que corresponde a una estrategia basada en ANN.

La figura 4.1 muestra el esquema general que siguen los dos tipos de estrategias que hemos utilizado en los análisis del presente trabajo. Se pueden ver claramente dos líneas principales de estrategias a seguir: una basada en CART y la otra basada en ANN, donde el producto final será la determinación del algoritmo que permita una eficiente modelización de la respuesta bajo el criterio de estabilidad estadística. Todos los análisis numéricos en esta fase se realizan considerando la raíz cuadrada del error cuadrático medio (RMSE) como estadístico de prueba.

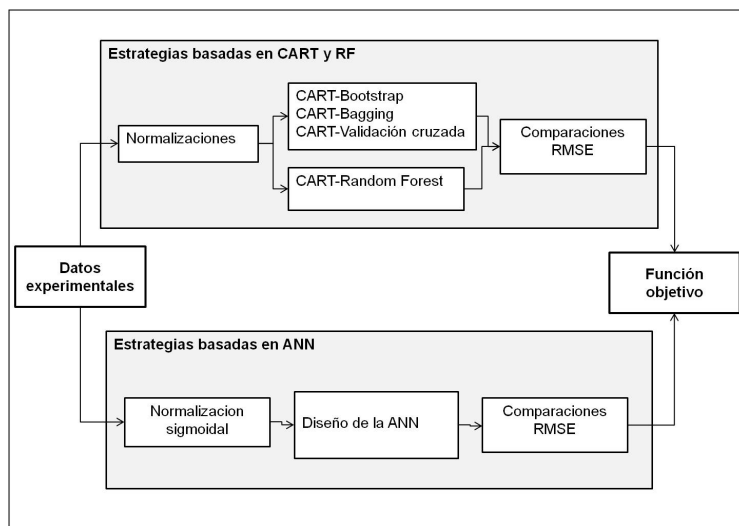


Figura 4.1: Esquema general de las estrategias basadas en CART y ANN

Con el fin de detallar los pasos de los algoritmos propuestos, conducentes al cálculo del RMSE como estadístico de prueba, hemos dividido

el presente capítulo en 3 secciones más. La sección (4.2) describe las estrategias basadas en CART, la cual presentamos por etapas: el punto (4.2.1) presenta y justifica las funciones empleadas en la normalización de los datos, seguidamente (punto 4.2.2) se muestra de manera general el desarrollo de los algoritmos predictivos que conducirán a la elección del algoritmo más robusto a ser contrastado con las estrategias basadas en ANN; seguidamente, para finalizar ésta sección (punto 4.2.3), se detallan los pasos a seguir para el desarrollo de cada algoritmo mencionado en el punto anterior. Por otra parte, la sección (4.3) es dedicada al desarrollo de la estrategia basada en ANN en concordancia con el trabajo presentado por Su y Chang (2000).

Para finalizar el presente capítulo y con el fin de realizar comparaciones numéricas exhaustivas, se consideran dos casos de ilustración en la sección (4.4). El primer caso corresponde a un estudio adoptado por Su y Chang (2000), el cual hemos usado como estudio piloto para determinar las virtudes y deficiencias de nuestros esquemas de modelización en contraste con las estrategias basadas en ANN. El segundo caso corresponde a un conjunto de datos con estructura más compleja, presentado por Chang (2005), el cual heredará el esquema resultante del primer caso con la finalidad de orientar la elección del esquema definitivo de modelización a emplear en la fase de optimización.

4.2. Estrategias basadas en CART y RF

Nuestra propuesta consiste en un esquema de trabajo cuyo producto final será la predicción de la respuesta para cada conjunto de valores de los factores de control. El entrenamiento de nuestros modelos predictivos, se

realizará inicialmente mediante estrategias que incluyen el uso de árboles de regresión con técnicas de remuestreo y posteriormente se incluye una estrategia basada en RF.

En el capítulo anterior se examinaron, de manera general, 3 métodos de remuestreo como estrategias que ayuden a minimizar la variabilidad propia de los árboles de regresión. Sin embargo, tales métodos poseen limitaciones que podrían afectar los rendimientos de los esquemas de modelización. En efecto, la validación cruzada es sensible al número de divisiones realizada a la muestra inicial, Hastie y otros (2009) recomiendan dividir tal muestra entre 5 y 10 grupos, lo cual sería arriesgado si dicha muestra no fuese lo suficientemente grande, sobre todo en el momento de fijar el número mínimo de observaciones por nodo y en los nodos terminales. Bootstrap, por su parte, realiza el remuestreo con reemplazamiento del conjunto inicial, quedando la muestra inicial como conjunto de prueba en el ajuste del árbol de regresión; por tanto las observaciones comunes entre todas las muestras podría concluir en problemas de sobreajuste. Finalmente, Bagging se presenta como el método más adecuado para tratar la variabilidad propia de los árboles de regresión ya que permite, mediante la combinación de diferentes árboles, captar la estructura compleja del conjunto de datos. Sin embargo, cada árbol generado es idénticamente distribuido al provenir de muestras bootstrap, entonces el promedio de b variables aleatorias independientes e idénticamente distribuidas, cada una con varianza σ^2 , tendrá varianza $\frac{1}{b}\sigma^2$, pero si dichas variables no son independientes entonces tendrán varianza $\left(\rho + \frac{1+\rho}{b}\right)\sigma^2$. Por tanto, en presencia de variables aleatorias no independientes el beneficio obtenido por el uso de promedios es limitado.

Pese a todo lo anterior, hemos decidido mantener en nuestra propuesta

de modelización algoritmos que combinen las bondades que ofrecen los métodos de remuestreo presentados anteriormente con las ventajas que ofrece CART en el reconocimiento de patrones. Todo ello con la intención de establecer una metodología amplia y que se ajuste a las necesidades y características de los posibles usuarios y sus datos. Sin embargo, no pasamos por alto las limitaciones que dichas combinaciones puedan poseer, por lo cual agregamos a nuestra propuesta metodológica una estrategia basada en RF como método que combina eficientemente a Bootstrap y Bagging.

4.2.1. Normalización de los datos

Las estrategias basadas en CART llevarán a cabo una normalización de los datos como una fase previa en el desarrollo de los modelos predictivos, lo que obedece a dos razones. En primera instancia, la normalización permite reducir la variabilidad propia del conjunto de datos, por lo que no sólo nos concentramos en realizar la respectiva normalización, sino que proporcionamos tres funciones distintas de normalización a fin de seleccionar la mejor (en términos de variabilidad). Por otra parte, nuestra propuesta pretende ser una alternativa en el uso de las ANN como técnica para el establecimiento de la relación input/output presentado en los trabajos de Su y Chang (2000); Chang (2005), por lo que a fines de comparaciones numéricas es vital la igualdad de escalas en el conjunto de datos.

En efecto, los datos serán normalizados alternativamente mediante tres funciones a fin de elegir la mejor normalización en virtud a dos criterios: la dispersión de los datos y el rendimiento general de los algoritmos. Consideraremos como mejor normalización aquella que minimice la variación propia del conjunto de datos, de forma que los métodos predictivos que la incorporen presenten mayor robustez.

La tabla 4.1 presenta las funciones de normalización a comparar. Nótese como $f_1(x)$ (mediante la definición de a y β) transforma el conjunto de datos combinando operaciones logarítmicas que garantizan la continuidad de $f_1(x)$ y a su vez proporcionan una concentración en el intervalo cerrado $[0,15; 0,85]$. Por su parte, $f_2(x)$ y $f_3(x)$ se presentan como una ligera modificación de la función sigmoideal: $f_2(x)$ mediante la expresión $x' = \frac{x-\text{mín}}{\text{máx}-\text{mín}}$ permite reducir la amplitud del conjunto de datos y $f_3(x)$, mediante la conocida tipificación $x' = \frac{x-\mu_x}{\sigma_x}$, permite concentrar los datos en base a su valor central, con el riesgo de estar fuertemente afectada por valores extremos.

Tabla 4.1: Funciones de normalización

$f_1(x) = \begin{cases} 0 & x = 0 \\ 1 - (0,85)^a & \text{en otros casos} \end{cases}$	$\text{donde } a = \left(\frac{x}{\text{mín}}\right)^\beta, \beta = -\left(\frac{2,4573}{\ln\left(\frac{\text{mín}}{\text{máx}}\right)}\right)$
$f_2(x) = \begin{cases} 0 & x = 0 \\ \frac{1}{1+\exp(-x')} & \text{en otros casos} \end{cases}$	$\text{donde } x' = \frac{x-\text{mín}}{\text{máx}-\text{mín}}$
$f_3(x) = \begin{cases} 0 & x = 0 \\ \frac{1}{1+\exp(-x')} & \text{en otros casos} \end{cases}$	$\text{donde } x' = \frac{x-\mu_x}{\sigma_x}$

4.2.2. Desarrollo y selección de los algoritmos para la modelización

A continuación se presentan los pasos para el análisis de los algoritmos predictivos, el cual hemos llamado *procedimiento* A_0 , ya que conducirá a

la elección del algoritmo A_p ($p=1,\dots,4$) más robusto y que posteriormente será contrastado con la estrategia basada en ANN.

Como paso inicial de tal procedimiento, además de las normalizaciones presentadas en el punto 4.2.3, es necesario el ajuste de los parámetros de CART ya que constituyen la base para el desarrollo de los algoritmos predictivos A_p que proponemos. Se determina el número mínimo de observaciones en cada nodo y el número mínimo de observaciones en los nodos terminales; los cuales llamaremos *minsplit* y *minbucket* respectivamente. Finalmente, se debe ajustar el parámetro de complejidad $cp = \alpha$, definido en la sección 3.2, el cual será estimado mediante validación cruzada en *10-fold*. Todo esto con la finalidad de diseñar un árbol complejo al menor coste posible de la tasa de error.

Procedimiento A_0 : Análisis de las estrategias basadas en CART

Inicialización: Hacer $i = 1$, $p = 1$.

Paso 1: Aplicar normalización $f_i(x)$ al conjunto de datos originales.

Paso 2: Ajuste de CART mediante la muestra completa normalizada para determinar los parámetros *minsplit*, *minbucket* y *cp*, así como su *RMSE*, el cual denotaremos como $RMSE_{tree}^i$.

Paso 3: Entrenar el algoritmo predictivo A_p .

Paso 4 : Calcular $RMSE_{A_p}^i$ de las predicciones obtenidas para el conjunto test con A_p . Si $p < 4$, hacer $p = p + 1$ y volver al paso 3. En otro caso ir al Paso 5.

Paso 5: Determinar $p_i = \arg \min_{i=1,2,3} (RMSE_{A_p}^i)$, es decir, A_{p_i} es el algoritmo que proporciona el menor RMSE con la normalización

$f_i(x)$. Si $i < 3$, hacer $i = i + 1$ y volver al Paso 1. En caso contrario, ir al Paso 6.

Paso 6: $\forall p = 1, 2, 3, 4$ determinar $f_p^* = \arg \min_{i=1,2,3} (RMSE_{A_p}^i)$, es decir, la normalización que proporciona el menor RMSE con el algoritmo A_p .

Desarrollo de los algoritmos predictivos

El objetivo del presente punto es el desarrollo *paso a paso* de cada algoritmo predictivo A_p ($p=1,\dots,4$) mencionado en el procedimiento A_0 del punto anterior, los cuales hemos codificado como sigue.

A_1 : PRED-T+VC. Algoritmo que combina CART con validación cruzada.

A_2 : PRED-T+Boot. Algoritmo que combina CART con Bootstrap.

A_3 : PRED-T+Bagg. Algoritmo que combina CART con Bagging.

A_4 : PRED-RF. Algoritmo basado en Random Forest.

La figura 4.2 muestra en detalle el esquema a seguir por los algoritmos basados en CART que proponemos. La parte exterior del recuadro sombreado ilustra el procedimiento A_0 presentado en el punto 4.2.2 y la parte interior ilustra el procedimiento de cada algoritmo A_p . Por tanto, el esquema se inicia con las normalizaciones $f_i(x)$ del conjunto de datos (presentadas en el punto 4.2.1) y el ajuste CART; dando paso al desarrollo de cada algoritmo A_p para finalizar en la comparación de tales algoritmos en términos de los RMSE obtenidos, lo que corresponde al paso 6 del procedimiento A_0 .

Vale la pena acotar que cada uno de éstos algoritmos será probado con cada función de normalización presentada en la tabla 4.1 con el fin de no sólo determinar el mejor algoritmo predictivo, sino optimizar la función de

normalización a considerar para el contraste con los resultados arrojados en el desarrollo del algoritmo en ANN. Todo lo anterior conducirá finalmente a la metodología propuesta para la modelización de la respuesta en el problema de diseño que estamos tratando para el caso univariante.

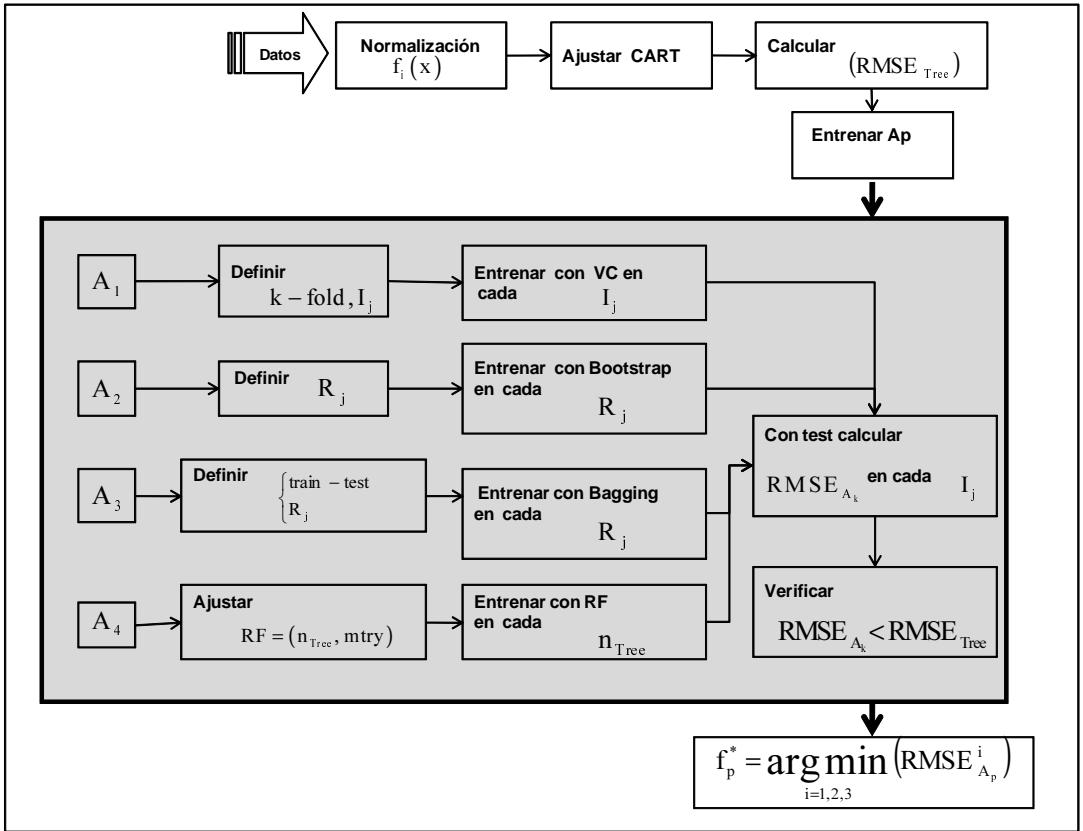


Figura 4.2: Estrategias basadas en CART

A continuación se detallan los pasos a seguir en el desarrollo de cada algoritmo predictivo A_p , ($p = 1, 2, 3, 4$) del paso 3 en el procedimiento A_0 , representados en el rectángulo sombreado de la figura anterior. Recordamos que todos estos algoritmos reciben como entrada los mismos valores de los

parámetros $minsplits$, $minbucket$, cp y $RMSE_{tree}^i$ obtenidos en el paso 2 del procedimiento A_0 .

Algoritmo A_1 :PRED-T+VC

Inicialización: Seleccionar el valor de k =número de subconjuntos en que se dividirá el conjunto de datos normalizados. Ir al Paso 1.

Paso 1: Dividir aleatoriamente la muestra original en k subconjuntos de igual tamaño $\{fold_1, \dots, fold_k\}$. Hacer $j = 1$ e ir al Paso 2.

Paso 2: Determinar el conjunto de entrenamiento y el conjunto de validación para j , definidos como $train_j = \{fold_1, \dots, fold_{j-1}, fold_{j+1}, \dots, fold_k\}$ $test_j = fold_j$, respectivamente. Ir al paso 3.

Paso 3: Entrenar el modelo CART definido en el Paso 2 del procedimiento A_0 con la muestra $train_j$. Obtener las predicciones de las respuestas para el conjunto $test_j$ mediante dicho árbol e ir al Paso 4.

Paso 4: Calcular el $RMSE_{T+CV}(j)$ obtenido con las respuestas predichas para el conjunto $test_j$. Si $RMSE_{T+CV}(j) < RMSE_{tree}$ ir al Paso 5. En otro caso, volver al paso 2 del procedimiento A_0 para obtener nuevos valores de los parámetros $minsplits$, $minbucket$, cp y $RMSE_{tree}^i$, hacer $j=1$ e ir al Paso 2.

Paso 5: Si $j < k$, hacer $j = j + 1$ e ir al Paso 2. En otro caso, calcular

$$RMSE_{T+CV} = \frac{\sum_{j=1}^k RMSE_{T+CV}(j)}{k}, \text{ y finalizar.}$$

Algoritmo A_2 :PRED-T+Boot

Inicialización: Seleccionar el valor de k =número de réplicas con reemplazamiento de la muestra original. Ir al Paso 1.

Paso 1: Elegir aleatoriamente y con reemplazamiento las m réplicas $\{R_1, \dots, R_m\}$ de la muestra original y definir los conjuntos complementarios de las réplicas en la muestra original $\{T_1, \dots, T_m\}$. Hacer $j = 1$ e ir al Paso 2.

Paso 2: Entrenar el modelo CART definido en el Paso 2 del procedimiento A_0 con la réplica R_j . Obtener las predicciones de las respuestas para el conjunto T_j mediante dicho árbol e ir al Paso 3.

Paso 3: Calcular el $RMSE_{T+Boot}(j)$ obtenido con las respuestas predichas para el conjunto T_j . Si $RMSE_{T+Boot}(j) < RMSE_{tree}^i$ ir al Paso 4. En otro caso, volver al paso 2 del procedimiento A_0 para obtener nuevos valores de los parámetros $minsplit, minbucket, cp$ y $RMSE_{tree}^i$, hacer $j=1$ e ir al Paso 2.

Paso 4: Si $j < k$, hacer $j = j + 1$ e ir al Paso 2. En otro caso, calcular $RMSE_{T+Boot} = \frac{\sum_{j=1}^m RMSE_{T+Boot}(j)}{m}$, y finalizar.

Algoritmo A_3 :PRED-T+Bagg

Inicialización: Seleccionar el valor de m =número de réplicas y n^* =tamaño muestral de cada réplica sin reemplazamiento de la muestra, ($m > n^*$). Ir al Paso 1.

Paso 1: Dividir aleatoriamente la muestra original en conjunto de entrenamiento (*train*) y conjunto de validación (*test*). Elegir aleatoriamente y sin reemplazamiento las m réplicas $\{R_1, \dots, R_m\}$ del conjunto (*train*). Hacer $j = 1$ e ir al Paso 2.

Paso 2: Entrenar el modelo CART definido en el Paso 2 del procedimiento A_0 con la réplica R_j . Obtener las predicciones de las respuestas para

el conjunto *test* mediante dicho árbol e ir al Paso 3.

Paso 3: Calcular el $RMSE_{T+Bagg}(j)$ obtenido con las respuestas predichas para el modelo j -ésimo del conjunto *test*. Si $RMSE_{T+Bagg}(j) < RMSE_{tree}^i$ ir al Paso 4. En otro caso, volver al paso 2 del procedimiento A_0 para obtener nuevos valores de los parámetros *minsplit*, *minbucket*, *cp* y $RMSE_{tree}^i$, hacer $j=1$ e ir al Paso 2.

Paso 4: Si $j < k$, hacer $j = j + 1$ e ir al Paso 2. En otro caso, calcular $RMSE_{T+Bagg} = \frac{\sum_{j=1}^m RMSE_{T+Bagg}(j)}{m}$, y finalizar.

Algoritmo A_4 :PRED-RF

Inicialización: Determinar los parámetros propios del Random Forest, como son el número de árboles a ensamblar (n_{tree}) y el número de variables de la muestra aleatoria que serán candidatos en cada división (*mtry*); este último parámetro debe ser optimizado mediante un estudio preliminar de la tasa de error OOB. Ir al Paso 1.

Paso 1: Dividir aleatoriamente la muestra original en conjunto de entrenamiento (*train*) y conjunto de validación (*test*). Ir al Paso 2.

Paso 2: Entrenar el modelo Random Forest (*RF*) con el conjunto *train*. Obtener las predicciones de las respuestas para el conjunto *test* mediante dicho modelo e ir al Paso 3.

Paso 3: Calcular el $RMSE_{RF}$ obtenido con las respuestas predichas para conjunto *test*. Si $RMSE_{RF} < RMSE_{tree}^i$ finalizar. En otro caso, volver al paso 2 del procedimiento A_0 para obtener nuevos valores

de los parámetros $minsplrit$, $minbucket$, cp y $RMSE_{tree}^i$, hacer $j=1$ e ir al Paso 2.

Para continuar con el esquema general presentado en la sección 4.1 e ilustrado en la figura 4.1 damos paso a las estrategias de predicción basadas en ANN, las cuales detallamos en la siguiente sección.

4.3. Estrategias basadas en ANN

Su y Chang (2000) y Chang (2005) presentan alternativas en la mejora del diseño de parámetros, las cuales proponen en su fase inicial emplear las ANN en la predicción de la respuesta. Dichos autores aseguran la buena adaptabilidad de las ANN a factores de tipo cuantitativo y cualitativo así como su fácil aplicación a problemas de ingeniería industrial. Por tal razón presentamos a continuación el algoritmo propuesto por tales autores, a fin de incorporarlo a nuestro análisis comparativo y en vista a que uno de nuestros objetivos en esta primera etapa, se basa en la búsqueda de estrategias robustas para la modelización de la respuesta aplicada a un problema de diseño de parámetros.

El algoritmo lo hemos codificado como $A_5:PRED-ANN_{xxx}$, donde la forma xxx se refiere a la arquitectura que conduzca al menor valor de RMSE. Recordemos que el objetivo en este punto es determinar la mejor arquitectura, en términos de RMSE; posteriormente incorporarla a los algoritmos predictivos y de entre todos determinar la estrategia más conveniente en la fase de modelización de la respuesta para el modelo univariante definido en (4.1). La figura 4.3 muestra el esquema de tal algoritmo, cuyo desarrollo se presenta a continuación.

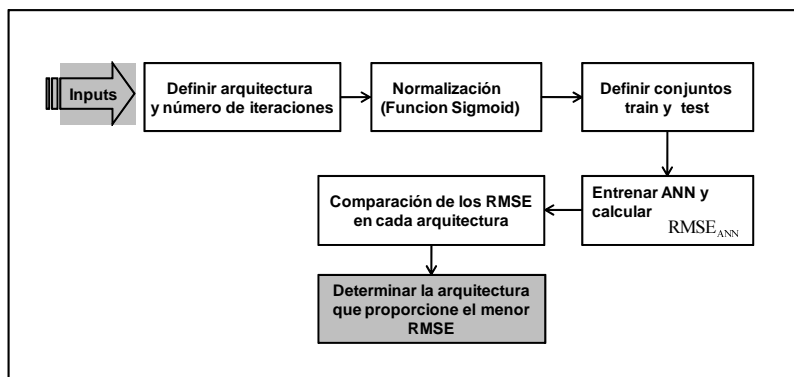


Figura 4.3: Estrategia basada en ANN

Algoritmo $A_5: PRED - ANNxxx$

Inicialización : Considerar la arquitectura de red xxx . Determinar el número n de iteraciones (*epochs*).

Paso 1: Normalizar el conjunto de datos mediante la función sigmoideal e ir al Paso 2.

Paso 2: Dividir la muestra original en conjuntos de entrenamiento (*train*) y validación (*test*). Ir al Paso 3.

Paso 3: Ajuste de la Red Neuronal bajo el esquema de Backpropagation mediante el conjunto de entrenamiento. Ir al paso 4.

Paso 4: Calcular $RMSE_{ANN}^{train}$ y $RMSE_{ANN}^{test}$ e ir al Paso 5.

Paso 5: Si procede, elegir un nuevo valor de n y volver al Paso 3. En otro caso, ir al paso 6.

Paso 6: Determinar la arquitectura que proporcione el menor RMSE para los diferentes valores de n .

4.4. Casos de ilustración

A continuación se ilustran los pasos de los algoritmos para la modelización de la función objetivo y se muestran comparaciones numéricas en cuanto a las virtudes y deficiencias de tales algoritmos, todo ello mediante 2 conjuntos de datos reales. Para llevar a cabo un análisis numérico exhaustivo, presentamos el primer caso de ilustración como un estudio piloto, en el que se realizan las comparaciones numéricas pertinentes de todos los esquemas presentados a lo largo del presente capítulo, incluyendo el algoritmo propuesto por Su y Chang (2000). Seguidamente, mediante el segundo caso de ilustración, se verifican las virtudes y deficiencias del algoritmo de modelización resultante del caso anterior en contraste con los resultados obtenidos por Chang (2005) en su trabajo.

Dividimos la presente sección en 3 partes. El apartado 4.4.1 presenta los principales parámetros que rigen todos los algoritmos de modelización así como los lenguajes de programación y librerías empleadas. El apartado 4.4.2 corresponde al primer caso de ilustración. Iniciamos mediante una breve descripción del conjunto de datos y continuamos con el estudio de las funciones de normalización mediante un análisis de la varianza (ANOVA) que nos permitirá elegir de manera más rigurosa la función a emplear en el establecimiento de las estrategias de modelización basadas en CART y RF. Este punto tiene como finalidad determinar el mejor algoritmo de modelización, en términos de RMSE, en contraste con el algoritmo basado en ANN presentado por Su y Chang (2000). En el apartado 4.4.3 se presenta un conjunto de datos con estructura más compleja, el cual será modelado mediante el algoritmo resultante en el análisis del caso anterior y será nuevamente contrastado con el algoritmo basado en ANN, pero esta vez considerando los resultados mostrados en el trabajo de Chang

(2005).

Todo lo anterior tiene como propósito determinar el esquema definitivo de modelización en nuestra propuesta metodológica para la mejora del diseño de parámetros en casos univariantes, así como establecer un punto de partida que nos permita generalizar nuestra propuesta a casos multivariantes.

4.4.1. Parámetros de control

Las diferentes técnicas estudiadas requieren previamente el ajuste de algunos parámetros de control. A continuación se enumeran dichos ajustes y el lenguaje de programación empleado en cada caso.

Estrategias basadas en CART:

- Lenguaje R, Librería rpart, versión 3.1-46 (Therneau y Atkinson, 2010).
- $\text{minsplit} = 10$, $\text{minbucket} = 2$, $\text{cp} = 0,00001$.

Algoritmo PRED-T+VC (Árboles de regresión + Validación cruzada)

- Lenguaje R, librería ipred, versión 0.8-8 (Peters y Hothorn, 2009).
- $10 - fold, I_j = n_{VC} = \{1000, 5000, 10000, 15000\}$.

Algoritmo PRED-T+Boot (Árboles de regresión + Bootstrap)

- Lenguaje R, librería ipred versión 0.8-8 (Peters y Hothorn, 2009).
- $R_j = n_{Boot} = \{1000, 5000, 10000, 15000\}$.

Algoritmo PRED-T+Bagg (Árboles de regresión + Bagging)

- Lenguaje R, librería `ipred` versión 0.8-8 (Peters y Hothorn, 2009).
- $R_j = n_{Boot} = \{1000, 5000, 10000, 15000\}$.
- Estimación del RMSE mediante out-of-bag (OOB).

Algoritmo PRED-RF

- Lenguaje R, librería `randomForest` versión 4.5-33 (Liaw y Matthew, 2009).
- $n_{Tree} = \{1000, 5000, 10000, 15000\}$, $mtry = 4$.
- Estimación del RMSE mediante OOB.
- No se establece criterio de poda entre los árboles.

Cabe resaltar que, en los algoritmos PRED-T+CV, PRED-T+Boot, PRED-T+Bagg y PRED-RF, se mantienen los mismos controles considerados en los árboles de regresión planteados al inicio de este apartado.

Estrategias basadas en ANN

- Software `Qnet2000` (2000).
- Redes Neuronales en Backpropagation.
- Normalización mediante la función sigmoid.
- Arquitecturas: $8-3-1, 8-4-1, 8-5-1, 8-6-1, 8-7-1, 8-8-1$ en las iteraciones $\{1000, 5000, 10000$ y 15000 respectivamente}

4.4.2. Caso 4.1

Descripción del conjunto de datos

El moldeo por inyección asistida por gas es una técnica que consiste en el llenado parcial, casi completo, de la cavidad del molde con un polímero convencional, seguido por la inyección de un gas inerte a presión. El proceso transcurre en tres etapas; en primer lugar se inyecta el polímero fundido, ocupando entre el 75 % y el 98 % del volumen de la cavidad del molde y tras un intervalo de tiempo, se inyecta el gas que empuja al polímero completando el llenado de la cavidad; la tercera y última etapa, denominada etapa de compactación con gas, se utiliza para contrarrestar la contracción que sufre el plástico al enfriarse. Esta técnica ofrece grandes ventajas frente al moldeo por inyección convencional en cuanto a la calidad del producto, donde el control de parámetros es un punto fundamental para llevar a cabo con éxito el punto final de dicho proceso.

El presente caso corresponde a un estudio adoptado por Su y Chang (2000) y corresponde a un proceso de moldeo por inyección asistida por gas, como el mencionado anteriormente. Los datos se presentan en una matriz ortogonal L_{18} con 8 factores de control $\mathbf{X} = \{A, B, C, D, E, F, G, H\}$ y 5 repeticiones (muestras), donde las respuestas y_i , ($i = 1, 2, \dots, 5$) representan la longitud en el canal de gas. El conjunto de datos se muestra en el anexo A, tabla 12.1. El objetivo de dicho estudio es determinar la combinación óptima del vector \mathbf{X} que minimice el valor de respuesta.

Los factores de control se definen como sigue:

A: Temperatura del molde	E: Presión del gas
B: Temperatura de fusión	F: Distancia del gas
C: Velocidad de inyección	G: Tiempo de retraso del gas
D: Tiempo de inyección de gas	H: Tiempo de presión constante

Considerando el enfoque del AS y en concordancia al trabajo presentado por Su y Chang (2000) a efecto de las comparaciones pertinentes, el conjunto de datos es dividido aleatoriamente en dos conjuntos. El conjunto de entrenamiento consta de 72 observaciones (80% de los datos) y el conjunto de prueba consta de 18 observaciones (20% de los datos).

Estudio de las funciones de normalización

La figura 4.4 muestra, mediante un gráfico de dispersión, la variabilidad del conjunto de datos considerando las funciones de normalización definidas en la tabla 4.1. Es fácil observar que es f_2 la función que concentra más las respuestas, lo que puede conducir a RMSE más bajos y por lo tanto un mejor ajuste de la función objetivo a establecer.

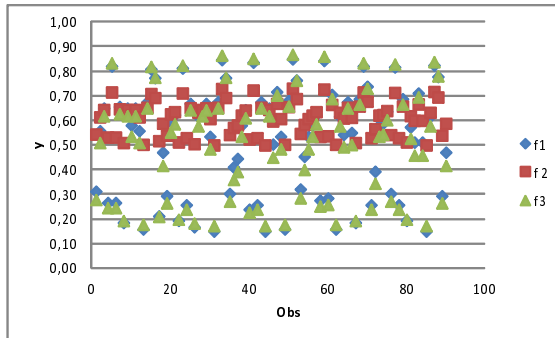


Figura 4.4: Funciones de normalización

Sin embargo, a efectos de mayor rigurosidad en la elección de la función

de normalización se realiza un ANOVA bajo un diseño completamente aleatorizado, el cual arroja diferencias altamente significativas ($p\text{-value} = 0,000137$) entre las funciones de normalización. Las medias y desviaciones estándar por función de normalización se muestran en la figura 4.5, donde si bien es cierto f_2 posee el mayor valor de media, (f_1 y f_2 poseen menor media por la presencia de valores extremos) también es cierto que su desviación estándar es muy inferior con respecto al resto de funciones; lo cual a efecto de los objetivos del presente estudio proporciona mayor estabilidad estadística en los resultados de los RMSE.

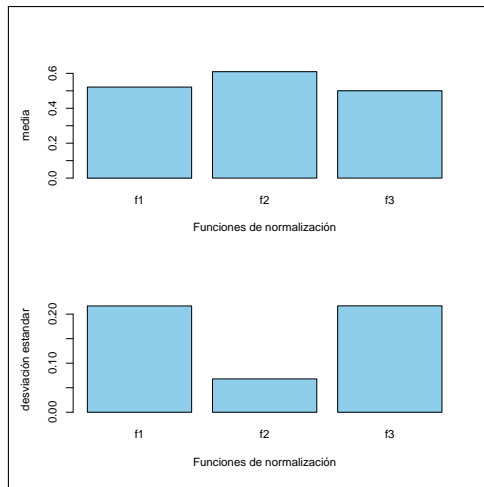


Figura 4.5: Media y Desviación estándar por función de normalización

Comparación de los modelos predictivos

A fines de realizar un análisis numérico exhaustivo, hemos dividido la presente sección en 3 partes. Una primera parte dedicada a los análisis correspondientes a las estrategias basadas en CART con el objetivo de determinar entre ellas la de mejor comportamiento. En la segunda parte, exploramos los resultados obtenidos mediante la estrategia basada en ANN

a fin de determinar la mejor arquitectura; todo esto en concordancia con la propuesta establecida por Su y Chang (2000). En la última parte, hacemos las respectivas comparaciones entre la mejor estrategia basada en CART y la mejor arquitectura de la estrategia PRED-ANN obtenidas en las secciones anteriores; todo ello con el fin de establecer finalmente una estrategia óptima para la modelización de la respuesta en el caso univariante. Cabe resaltar que en todas las comparaciones realizadas se ha establecido como estadístico de prueba al RMSE.

Resultados numéricos de los algoritmos predictivos basados en CART

Como punto de inicio, tras establecer la función de normalización, es necesario ajustar el árbol de regresión para ser combinado con las diferentes técnicas de remuestreo aquí empleadas y con el RF a desarrollar. La figura 4.6 muestra el estudio del parámetro de complejidad (cp) en contraste con el error relativo. Aquí el objetivo se concentra en elegir el árbol más complejo posible (mayor número de divisiones) al menor coste de clasificación o tasa de error. Esto es, si se toma el mínimo valor cp , la penalidad por tener un árbol con un número de nodos terminales menor o igual al árbol más complejo será mínima (Breiman y otros, 1984). Entonces en dicha figura se puede observar que la tasa de error se estabiliza a partir del árbol de tamaño 8, entonces, podemos elegir un tamaño 15 ($size = 15$) con un $cp = 0,00016$.

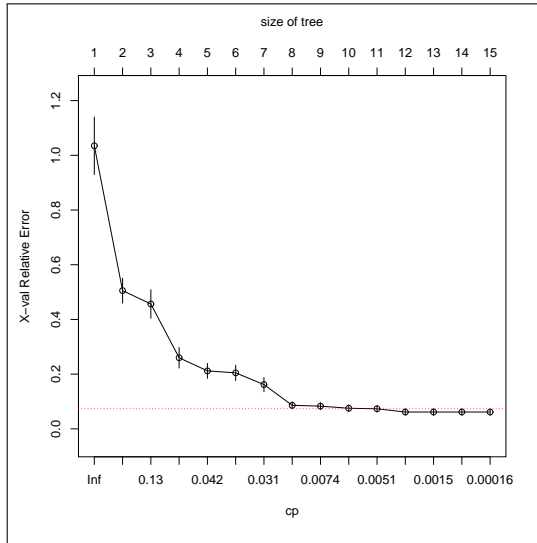


Figura 4.6: Ajuste del parámetro de complejidad (cp). Caso 4.1

Los ajustes de los parámetros establecidos anteriormente dan origen al árbol base para el desarrollo de los algoritmos de modelización diseñados en el presente estudio, el cual se presenta en la figura 4.9, ubicado al final de la presente sección.

Para finalizar el ajuste de los parámetros, solo resta hacerlo con m_{try} , definido por Breiman (2001) como $m_{try} = \frac{p}{3}$, siendo p el número de predictores en el conjunto de datos. En nuestro caso $m_{try} = \frac{8}{3} = 2,67 \approx 3$. Ahora bien, una de las potentes ventajas del algoritmo RF es que permite monitorizar éste parámetro mediante la tasa de error OOB. A tal fin se presenta la figura 4.7, donde se muestra el comportamiento de dicha tasa para distintos valores de m_{try} , fijándose de esta manera el parámetro $m_{try} = 4$.

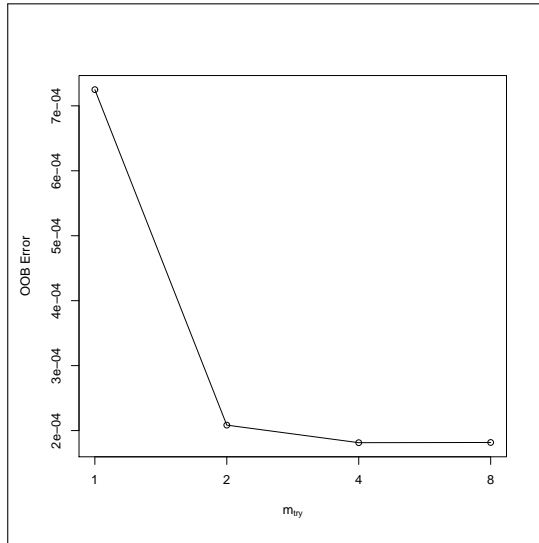


Figura 4.7: Ajuste del parámetro m_{try} mediante la tasa de error OOB.
Caso 4.1

Para finalizar este apartado, se presenta la figura 4.8, donde se muestran los rendimientos de RMSE de todas las estrategias basadas en CART. Se puede observar que los valores más altos de RMSE son arrojados por el algoritmo PRED-T+Boot seguidos por PRED-T+VC y PRED-T+Bagg; de éstos, note cómo es PRED-T+VC quien experimenta más cambios con respecto a los diferentes tamaños de iteración lo que se traduce en inestabilidad del método. Por su parte, es PRED-RF quien no solo presenta los menores valores de RMSE, sino que podemos decir que se mantiene estable en todas las iteraciones. Esto se traduce en ahorro de recursos computacionales, ya que basta con ensamblar 1000 árboles para obtener el menor RMSE.

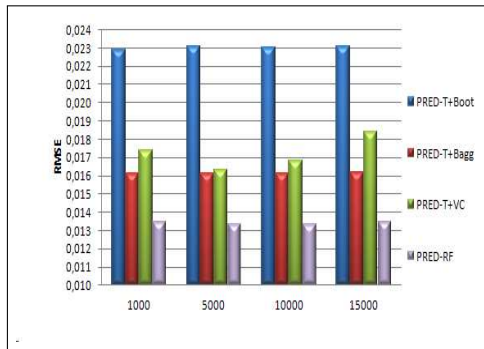


Figura 4.8: Comparación del RMSE de las estrategias basadas en CART.
Caso 4.1

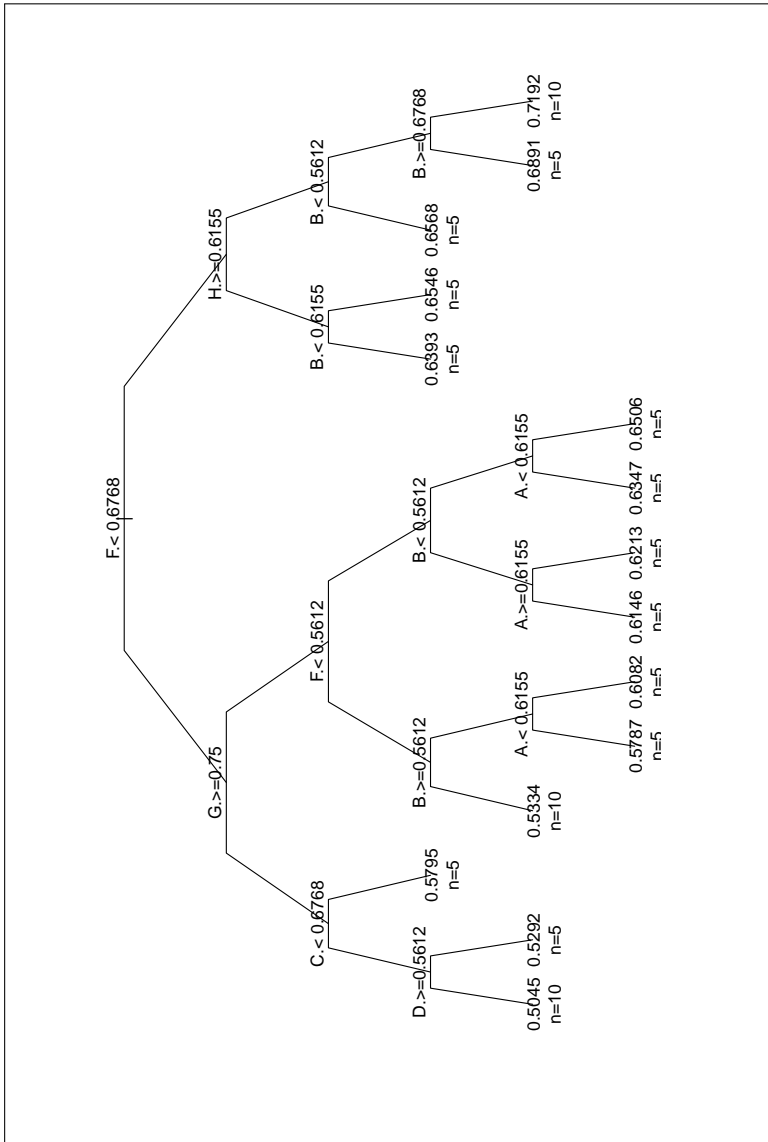


Figura 4.9: Árbol de regresión ajustado.

Resultados numéricos de los algoritmos predictivos basados en ANN

La figura 4.10 muestra los rendimientos de RMSE considerando todas las arquitecturas propuestas por Su y Chang (2000). Se pueden notar dos extremos bien marcados para todas las arquitecturas: los valores más altos de RMSE se obtienen en la iteración 1000 y los más bajos en la iteración 15000, lo que se traduce en coste computacional. En cuanto a la inestabilidad de la red, se puede ver cómo se mantienen estables los valores para cada iteración con excepción de la iteración inicial, lo que evidentemente muestra a la red estable sólo a partir de la iteración 5000. Su y Chang (2000) en su estudio, recomiendan como mejor arquitectura la 8 – 5 – 1 por lo que, a nuestros fines comparativos, tomaremos dicha arquitectura para la comparación de los resultados obtenidos en la sección siguiente.

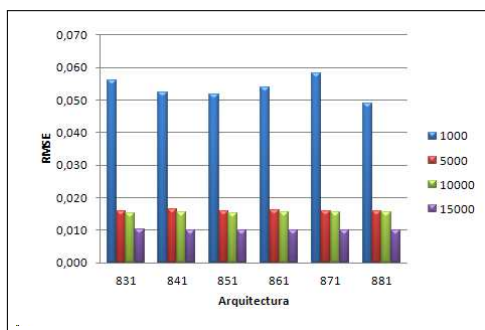


Figura 4.10: Comparación del RMSE de las estrategias basadas en ANN.
Caso 4.1

Comparaciones de los modelos predictivos basados en CART con PRED-ANN

Las comparaciones numéricas correspondientes a la presente sección la hemos realizado considerando dos enfoques bien claros. Inicialmente realizamos las comparaciones en concordancia con todos los anteriores análisis, es decir, manteniendo una escala normalizada. Posteriormente, hemos devuelto los correspondientes outputs a la escala inicial a fin de determinar más claramente las diferencias reales de las estrategias, así como las ventajas y desventajas que cada una aporta al proceso de modelización univariante.

Datos normalizados

A continuación se presentan los rendimientos de RMSE en todas las estrategias consideradas en el presente estudio, donde a efectos de notación llamaremos a la estrategia basada en ANN en la arquitectura $8 - 5 - 1$ como PRED-ANN851. Así pues, la figura 4.11 evidencia la inestabilidad de PRED-ANN851 mostrada en la figura 4.10 con respecto al resto de las estrategias bajo estudio. En efecto, si consideramos la iteración $n = 1000$ evidentemente PRED-ANN851 se muestra totalmente ineficiente dado al gran valor de RMSE alcanzado. Si consideramos ahora la iteración $n = 15000$ es precisamente PRED-ANN851 quien presenta los valores mas bajos de RMSE pero con valores muy cercanos a los alcanzados por PRED-RF, por lo que se puede decir, en términos del número de iteraciones que es PRED-RF quien permanece como el mejor método en cuanto al consumo de recursos computacionales.

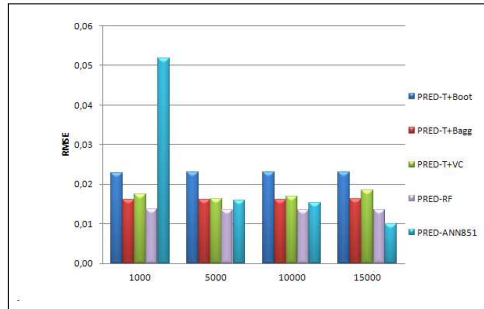


Figura 4.11: Comparación de los rendimientos de RMSE para todas las estrategias. Caso 4.1

Para cuantificar lo anteriormente dicho, se define la medida $\mathbf{P\%}$ como el porcentaje de *progreso* que implica el uso del algoritmo PRED-RF frente al algoritmo PRED-ANN851 en términos de RMSE. Los resultados se muestran en la tabla 4.2, donde se puede observar que a tan solo 1000 iteraciones PRED-RF presenta un progreso del 73.86%. También se puede verificar la estabilidad de los valores RMSE en PRED-RF para todas las iteraciones, lo cual no sucede para PRED-ANN851, quien mejora sus valores al aumentar el número de iteraciones al punto de alcanzar un progreso del 36.14% frente a PRED-RF cuando se ejecutan 15000 iteraciones.

Tabla 4.2: Progreso de los valores de RMSE de PRED-RF frente a PRED-ANN851. Caso 4.1

n	RMSE		
	PRED-RF	PRED-ANN851	$\mathbf{P\%}$
1000	0,0135	0,0517	73,86
5000	0,0134	0,0158	15,45
10000	0,0134	0,0153	12,88
15000	0,0135	0,0099	-36,14

Hasta ahora, y como primera conclusión que podría ayudar a concentrar la atención en el logro de los objetivos del presente trabajo, se puede decir que las estrategias basadas en ANN (en la arquitectura 8-5-1) y PRED-RF son los algoritmos que aportan los menores valores de RMSE, y por lo tanto las mejores estrategias en la determinación de la función objetivo, el cual constituye la meta de esta primera fase de nuestro estudio. Por tal razón, resulta interesante examinar de nuevo y más detenidamente dichas técnicas a fin de determinar sus diferencias. Entonces, la presente sección se enfoca ahora en la comparación más detallada de los resultados, considerando dos criterios fundamentales, como son la robustez estadística y el coste computacional. En efecto, la robustez estadística es determinada mediante la estabilidad de las medidas de RMSE en las diferentes iteraciones, tanto en el conjunto de entrenamiento como en el conjunto de prueba. Por su parte, el coste computacional es determinado mediante el número total de iteraciones necesarias para el logro del menor RMSE; cabe recordar que, en el caso de la modelización con PRED-RF, las iteraciones se traducen en el número total de árboles ensamblados.

Comparaciones de RMSE en escala inicial

Se toma una nueva muestra, esta vez sin reemplazamiento, considerando nuevamente el conjunto de entrenamiento con 72 observaciones y el conjunto de prueba con 18 observaciones; todo esto en correspondencia con el trabajo de Su y Chang (2000). Se realiza el estudio comparativo de los rendimientos de RMSE considerando los inputs normalizados y los outputs en su escala inicial (no normalizados). Cabe resaltar que, en cuanto al algoritmo PRED-RF se mantienen los parámetros del árbol de regresión inicial fijados a inicios del presente apartado. En cuanto a la estrategia basada en ANN, se considera la arquitectura 8 – 5 – 1.

Las comparaciones de los RMSE de PRED-RF, en los conjuntos de entrenamiento y prueba, se muestran en la figura 4.12, donde la similitud entre las columnas de entrenamiento y prueba, en todas las iteraciones, nos refleja la falta de sobreajuste del método; lo cual se traduce en rigurosidad y estabilidad estadística.

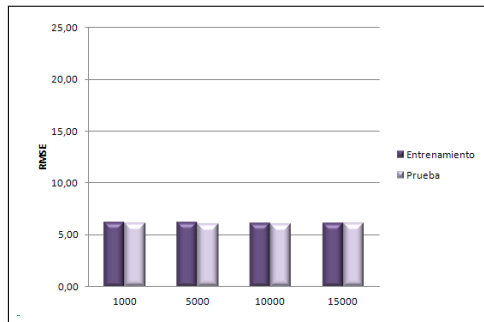


Figura 4.12: Comparación RMSE de PRED-RF en conjunto de entrenamiento y prueba. Caso 4.1

En cuanto a las comparaciones de los RMSE de PRED-ANN851, en los conjuntos de entrenamiento y prueba, son presentados en la figura 4.13. Ahora encontramos una situación diferente a la presentada en la figura 4.12. A simple vista se puede notar la gran diferencia, en términos de RMSE, de la iteración 1000 con respecto a las restantes, lo que nuevamente ratifica los resultados obtenidos al inicio del presente apartado (figura 4.10), donde se apreciaba que las estrategias basadas en ANN, en cualquier arquitectura, mejoran sus valores de RMSE a partir de 5000 iteraciones. Así mismo, se observan diferencias entre las columnas de entrenamiento y prueba en cada iteración. Los valores alcanzados por los conjuntos de prueba se mantienen superiores a los alcanzados por los conjuntos de entrenamiento, lo que nos hace pensar en posibles problemas de sobreajuste.

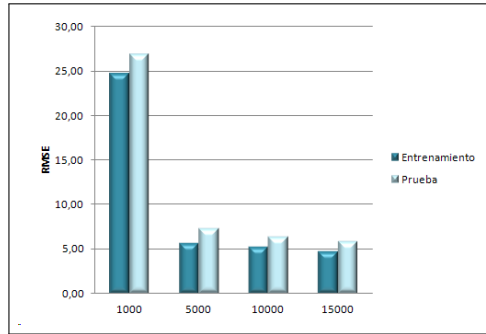


Figura 4.13: Comparación RMSE de PRED-ANN851 en conjunto de entrenamiento y prueba. Caso 4.1

Si presentamos los resultados de los algoritmos a fin de estudiar su comportamiento en los conjuntos de entrenamiento y prueba por separado, sigue confirmándose que PRED-RF es el algoritmo más estable, mientras que PRED-ANN851 se muestra completamente ineficiente en la iteración 1000 y sólo logra estabilizarse a partir de 5000 iteraciones, habiendo además una presencia clara del sobreajuste. Todo lo anterior se refleja en las figuras 4.14(a) y 4.14(b).

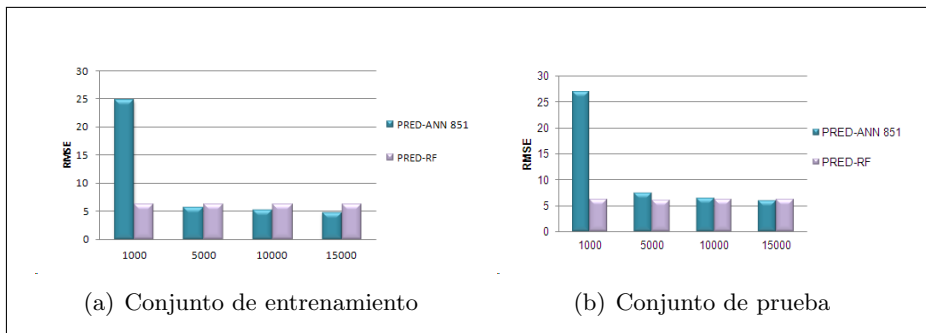


Figura 4.14: Comparación RMSE de PRED-RF y PRED-ANN851. Caso 4.1

En la figura 4.14(a) referida al conjunto de entrenamiento, se puede notar la cercanía entre los rendimientos de los respectivos RMSE sólo a partir de 5000 iteraciones, incluso se podría considerar a *PRED – ANN851* con los mejores resultados en el establecimiento de nuestra función objetivo. Pero, dicha conclusión no es tan firme si observamos la figura 4.14(b), referida al conjunto de prueba, donde los valores de RMSE permanecen un tanto mayores en el caso de *PRED-ANN851* con respecto a *PRED-RF*, lo que se puede traducir en problemas de sobreajuste. Tales gráficos nos orientan visualmente en cuanto a las diferencias y similitudes entre ambos algoritmos, pero en vista de que las mismas se minimizan a partir de 5000 iteraciones y que pasar de 5000 a 10000 ó 15000 iteraciones implica gastos de recursos computacionales, presentamos la tabla 4.3, donde se pueden apreciar las diferencias cuantitativas más claramente, así como establecer una medida de comparación numérica que permita medir la mejoría, en términos de RMSE, que supone el uso de *PRED-RF* frente a *PRED-ANN851*.

La tabla 4.3, presenta las diferencias numéricas existentes entre *PRED-RF* y *PRED-ANN* en la arquitectura 851, así como una medida de progreso **P%** definida anteriormente como el porcentaje de *progreso* que implica el uso del algoritmo *PRED-RF* frente al algoritmo *PRED-ANN851* en términos de RMSE. Vale la pena detallar lo que sucede en las iteraciones extremas, es decir, las iteraciones 1000 y 15000. Se puede ver que con sólo 1000 iteraciones se alcanzan mejoras del 75% y 77% de *PRED-RF* sobre *PRED-ANN851* en los conjuntos de entrenamiento y prueba respectivamente; lo que a efectos del presente trabajo se traduce en eficiencia estadística y ahorro de recursos computacionales cuando se emplea el algoritmo *PRED-RF*.

Por su parte PRED-ANN851 alcanza rendimientos del 28 % y 4 % en los conjuntos de entrenamiento y prueba respectivamente en su nivel más alto de iteraciones (15000). La enorme diferencia entre tales rendimientos nuevamente evidencia la presencia del sobreajuste en términos RMSE, como se ha venido observando anteriormente. Además, si olvidamos el tema del sobreajuste, podemos observar que el algoritmo PRED-ANN851 requiere de 1500 iteraciones para alcanzar buenos valores de RMSE, lo que implica un gran consumo de recursos computacionales y por lo tanto aumentos en los tiempos de ejecución del algoritmo.

Tabla 4.3: Progreso de los valores de RMSE de PRED-RF frente a PRED-ANN, en conjuntos de entrenamiento y prueba. Caso 4.1

n	Entrenamiento			Prueba		
	PRED-RF	PRED-ANN	P %	PRED-RF	PRED-ANN	P %
1000	6,17	24,75	75,07	6,13	26,94	77,24
5000	6,16	5,69	-8,13	6,00	7,35	18,40
10000	6,15	5,21	-18,06	6,03	6,42	5,93
15000	6,14	4,78	-28,44	6,14	5,92	-3,73

En virtud de las buenas características teóricas que encierra RF en el reconocimiento de patrones y lo observado en todo el análisis numérico anterior, hemos seleccionado al algoritmo PRED-RF como nuestro esquema de modelización y será nuestro punto de partida para el siguiente caso de ilustración el cual, mediante una estructura de datos más compleja, será nuevamente comparado con los resultados mostrados en el trabajo de Chang (2005), quien basa su modelización en ANN. Todo ello con el propósito de determinar de una manera más exhaustiva el esquema definitivo de modelización en nuestra propuesta metodológica para la mejora del diseño de parámetros.

4.4.3. Caso 4.2

Descripción del conjunto de datos

El presente caso corresponde a un estudio adoptado por Chang (2005) referido a un proceso de encapsulado conocido como TAP (Tape Automated Bonded). Este método de conexión eléctrica de chips consiste en la fabricación de una lámina de polímero sobre la que se imprimen líneas de cobre mediante un proceso de deposición. La conexión o soldado se realiza simultáneamente para todos los *pads* del chip y después para los del encapsulado. Lo anterior implica que tal método requiere la fabricación específica de las interconexiones de la lámina para un circuito en concreto, por lo tanto especificar los elementos que intervienen es de vital importancia para llevar a cabo el proceso. Para más información sobre el proceso TAP consulte Rubio y otros (2000).

El experimento es conducido bajo 6 factores de control y un factor de ruido, dispuestos en una matriz L_{18} . El conjunto de datos se presenta en el anexo A. La respuesta corresponde al grosor del patron *epoxy*, donde la característica de calidad es del tipo estático NTB y el objetivo consiste en determinar los niveles de los factores control para hacer la respuesta cercana al valor 0.2285mm.

Los factores de control considerados son el calibre de la aguja (en pulgadas), velocidad (pulgadas/seg), línea de espaciado (en pulgadas), presión, altura de la punta (en pulgadas) y aceleración en (pulgadas/ seg^2); denotados como A, B, C, D, E y F respectivamente.

Implementación del algoritmo PRED-RF

Considerando los resultados obtenidos en el análisis numérico del caso de ilustración 4.1, realizamos la fase de modelización bajo el algoritmo PRED-RF, considerando la normalización f_2 de la tabla 4.4 solamente para factores A, D y F, dado que los restantes se encuentran en escala (0,1). Se adoptan además los parámetros de control de PRED-RF mostrados en la sección (4.4.1), a excepción del parámetro m_{try} dado que estamos en presencia de una base de datos diferente. El ajuste de dicho parámetro se muestra en la figura 4.15.

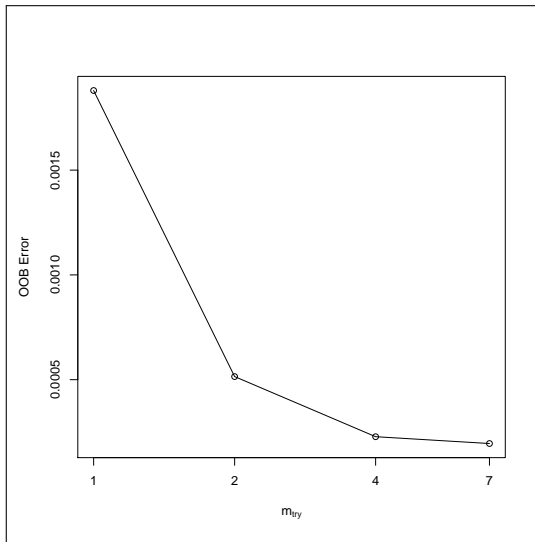


Figura 4.15: Ajuste del parámetro m_{try} mediante la tasa de error OOB. Caso 4.2

Se divide el conjunto de datos aleatoriamente en entrenamiento y prueba, 144 y 36 observaciones respectivamente, ya que posteriormente nuestros resultados serán comparados con los resultados obtenidos por Chang (2005) en su esquema de modelaje basado en ANN. El esquema de modelización

bajo el algoritmo PRED-RF queda estructurado como sigue.

Algoritmo PRED-RF

Inicialización: Aplicar normalización f_2 a los factores A, D y F.

- Ajuste CART: Hacer $minsplitleft = 10$, $minbucket = 2$ y $cp = 0,00001$.
- Ajuste RF: $n_{tree} = 1000$ y $mtry = 7$.
- Ir al Paso 1.

Paso 1: Dividir aleatoriamente la muestra original en conjunto de entrenamiento (144 obs.) y conjunto de prueba (36 obs.). Ir al Paso 2.

Paso 2: Entrenar RF con el conjunto de entrenamiento. Obtener las predicciones de las respuestas para el conjunto prueba. Ir al Paso 3.

Paso 3: Calcular el $RMSE_{RF}$ obtenido con las respuestas predichas para conjunto *test*. Si $RMSE_{RF} < RMSE_{tree}$, finalizar; en otro caso, volver al paso de inicialización para obtener nuevos valores de los parámetros $minsplitleft, minbucket, cp$ y $RMSE_{tree}^i$.

Comparación del algoritmo PRED-RF con ANN

La figura 4.16 compara los resultados de RMSE de PRED-RF y ANN para el presente caso de estudio, donde es evidente la reducción del RMSE con el empleo de PRED-RF. Acotamos que esta vez no hemos ejecutado el algoritmo PRED-ANN, sino que asumimos los resultados mostrados por Chang (2005) en la mejor arquitectura.

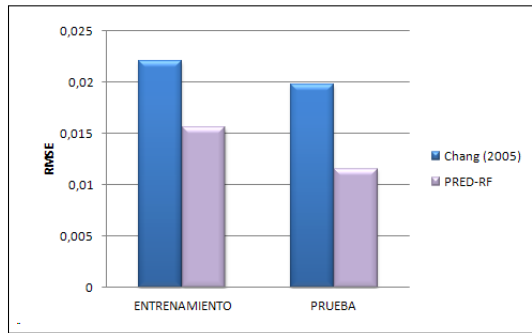


Figura 4.16: Comparación RMSE de PRED-RF con ANN en conjunto de entrenamiento y prueba. Caso 4.2

La tabla 4.4 cuantifica las diferencias vistas en la figura 4.16 a través de la medida de progreso $\mathbf{P}\%$, definida anteriormente como el porcentaje de *progreso* que implica el uso del algoritmo PRED-RF frente al algoritmo ANN en términos de RMSE. Nuevamente, pero con un conjunto de datos diferente, se observan notorias diferencias en cuanto a los rendimientos de RMSE, donde destacamos principalmente la medida $\mathbf{P}\% = 41\%$ en el conjunto de prueba, evidenciando la eficiencia de nuestro algoritmo PRED-RF en tan solo 1000 iteraciones frente a las 15000 necesarias por las ANN.

Tabla 4.4: Progreso de los valores de RMSE de PRED-RF frente a ANN, en conjuntos de entrenamiento y prueba. Caso 4.2

	Entrenamiento	Prueba	n
Chang (2005)	0.0221	0.0198	15000
PRED-RF	0.0156	0.0116	1000
$\mathbf{P}\%$	29.41	41.71	

Capítulo 5

Modelización para casos multivariantes

El presente capítulo se desarrolla como una generalización del capítulo anterior al caso multivariante. Proponemos métricas de modelaje, mediante el estudio de distintas funciones, que concentran las múltiples características de calidad (respuestas) por cada combinación de parámetros y que bajo una secuencia de árboles de regresión univariante tendrán como producto final las estimaciones de dichas respuestas. A dicha propuesta la hemos bautizado bajo el nombre de *Multivariate Random Forest* (MRF, por sus siglas en inglés).

Desarrollamos en primera instancia un esquema general de nuestra propuesta de modelización multivariante (sección 5.1) y posteriormente en la sección (5.2) se detallan paso a paso los elementos que intervienen en dicho esquema. Vale la pena acotar que en la sección (5.3) se mencionan brevemente las propuestas dadas por Chang (2008); Chang y Chen (2011), basadas en ANN, a fines comparativos. Finalizamos el capítulo con la

sección (5.4) presentando dos casos de ilustración que nos permiten validar nuestra propuesta y determinar sus fortalezas y debilidades

5.1. Esquema general de modelización

Suponemos un problema dinámico multi-respuesta definido como:

$$\mathbf{Y} = f_i(M_j, \mathbf{X}_k, Z_l) + \varepsilon_{ijkl} \quad (5.1)$$

donde $f_i(M_j, \mathbf{X}_k, Z_l)$ denota la función entre el $ijkl$ -ésimo vector respuesta \mathbf{Y} y el correspondiente j -ésimo nivel del factor señal, la k -ésima combinación del vector de control y el l -ésimo nivel del factor de ruido. Finalmente ε_{ijkl} representa un término de error. Nuestro modelo consta de r respuestas (outputs), s señales de ajuste, p factores control y n factores de ruido. Por tanto, la i -ésima respuesta y_{ijkl} es determinada por el conjunto de señales de ajuste $M_j = (M_1, \dots, M_s)$, el vector de factores control $\mathbf{X}_k = (x_1, \dots, x_p)$ y el conjunto de factores de ruido $Z_l = (z_1, \dots, z_n)$.

Entonces, como una generalización del caso univariante, presentamos nuestra propuesta en la etapa de modelización basada en el diseño de MRF como fase inicial en la solución de dicho problema y como una alternativa a los trabajos presentados por Chang (2006, 2008) y más recientemente Chang y Chen (2011). Cabe recordar que el algoritmo RF establecido por Breiman (2001) contempla sólo el caso de una variable respuesta (univariante). Hemos adaptado dicho algoritmo al caso de múltiples variables respuestas a fin de incorporarlo a nuestra propuesta de modelización, como se detallará posteriormente.

En sintonía con la figura 4.1 presentamos la figura 5.1, la cual muestra

el esquema general de modelización para el caso de múltiples respuestas. Acotamos que, con fines comparativos, nuevamente incorporamos la propuesta de Chang (2008) en nuestro trabajo bajo el mismo conjunto de datos, pero en este caso no hemos realizado las pruebas numéricas con respecto a ANN sino que hemos asumido los resultados presentados en su trabajo. Las comparaciones numéricas se realizan utilizando la raíz cuadrada del error cuadrático medio (RMSE) como estadístico de prueba.

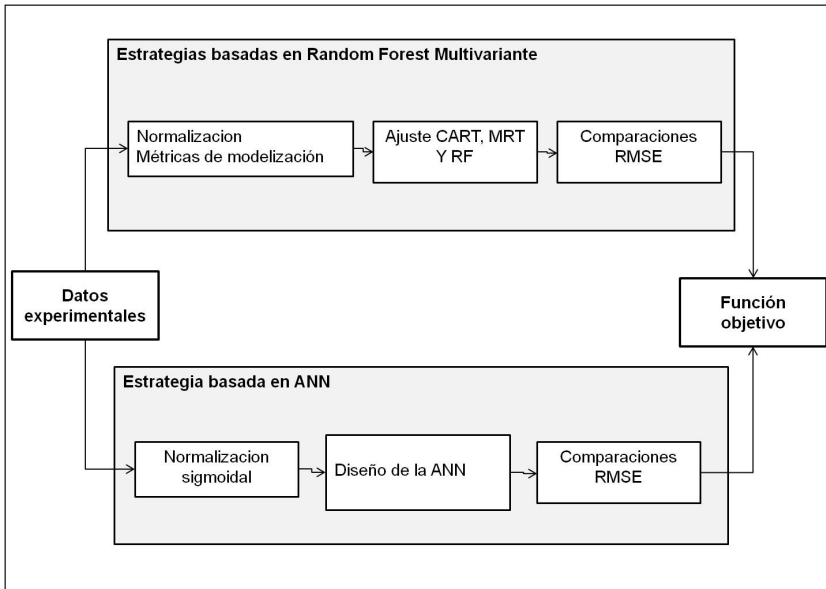


Figura 5.1: Esquema general de las estrategias basadas en MRF y ANN

El presente capítulo lo hemos dividido en tres secciones. La sección (5.2) corresponde a nuestra propuesta de modelización al caso multivariante en dos etapas, a saber: el apartado (5.2.1) corresponde a la preparación previa del conjunto de datos mediante la normalización y el diseño de distintas métricas para la unificación del vector de respuestas \mathbf{Y} , como estrategia para el diseño de MRF. El apartado (5.2.2) muestra los pasos para el desarrollo y la selección de los algoritmos de modelización basados

en MRF bajo las métricas diseñadas en el punto anterior, donde el producto final será la determinación del algoritmo de modelización que proporcione el menor RMSE como estadístico de prueba. La sección (5.3) explica brevemente la propuesta de modelización mostrada en los trabajos de Chang (2006, 2008) a efecto de las comparaciones numéricas posteriores. Finalmente, en la sección (5.4) se presentan dos casos de aplicación, con el fin de ilustrar las secciones anteriores mediante dos conjuntos de datos diferentes y realizar las respectivas comparaciones numéricas en cuanto a las virtudes y deficiencias de nuestra propuesta para el caso multivariante.

5.2. Estrategia basada en Random Forest multi-variante

Nuestra propuesta consiste en un esquema de trabajo cuyo producto final será la predicción de las respuestas para cada conjunto de valores de los factores de control. El entrenamiento de nuestro modelo predictivo se realiza mediante estrategias que incluyen el ajuste de árboles de regresión, funciones de normalización y métricas de modelaje para el diseño del MRF. Vale la pena resaltar que ajustamos un árbol de regresión multivariante (MRT) al conjunto de datos normalizados y determinamos su RMSE con la finalidad de establecer un valor límite para el desarrollo de nuestros MRF.

5.2.1. Normalización y métricas de modelaje

Nuestra propuesta ahora consta de dos fases previas en la preparación del conjunto de datos. En primera instancia, análogamente al caso univariante, proponemos la normalización del conjunto de datos a fin de minimizar su variabilidad interna y establecer similitud de escalas para las posteriores comparaciones numéricas. Así pues, en virtud a los buenos resultados

obtenidos hasta ahora, establecemos como función de normalización a f_2 definida como una leve modificación a la función sigmoideal y presentada en la tabla 4.1.

Seguidamente, se propone la unificación del vector de respuestas bajo métricas de modelaje definidas por el estudio de distintas funciones. Tales métricas concentran las múltiples características de calidad (respuestas) por cada combinación de parámetros que, bajo una secuencia de árboles de regresión univariante y la asignación de *pesos*, tendrán como producto final las estimaciones de tales respuestas. Vale la pena acotar que proponemos el estudio de diversas funciones en esta fase en virtud de los posibles efectos cuadráticos existentes en \mathbf{Y} . La tabla 5.1 presenta tales funciones, suponiendo un vector de respuestas $Y = (y_1, \dots, y_r)$ para cada m -ésima observación.

Tabla 5.1: Métricas de modelaje para los algoritmos MRF

$P_1(Y) = \sum_{i=1}^r y_i$	$P_5(Y) = \frac{1}{r} \sum_{i=1}^r y_i^2$
$P_2(Y) = \left(\prod_{i=1}^r y_i \right)^{\frac{1}{r}}$	$P_6(Y) = \frac{1}{r} \sum_{i=1}^r \exp(y_i)$
$P_3(Y) = \left(\prod_{i=1}^r y_i^{-1} \right)^{\frac{1}{r}}$	$P_7(Y) = \frac{1}{r} \sum_{i < s} \frac{y_i - \mu_s}{\sigma_s}$
$P_4(Y) = \left(\prod_{i=1}^r y_i^2 \right)^{\frac{1}{r}}$	

Acotamos que dichas funciones darán lugar a los algoritmos MRF_{P_p} ($p = 1, \dots, 7$) que se desarrollan a continuación, los cuales se inician con el ajuste del MRT y el cálculo de su correspondiente $RMSE_{MRT}$, como medida límite de nuestros algoritmos de predicción, y finalizar con la selección del algoritmo que proporcione el menor RMSE y por ende la mayor robustez en el proceso de modelización para nuestro caso de estudio.

5.2.2. Desarrollo y selección de los algoritmos para la modelización

Los árboles de regresión (CART) constituyen la base para el desarrollo de los algoritmos MRF_{P_p} , por tanto es necesario el ajuste de los parámetros $minsplitlet$, $minbucket$ y cp con el fin de determinar árboles complejos al menor coste posible de la tasa de error; así mismo se determina el correspondiente $MRSE$, el cual llamaremos $MRSE_{Tree_{P_p}}$. Seguidamente, se deben ajustar los parámetros propios de MRF, es decir, el número de árboles a ensamblar (n_{tree}) y el número de variables de la muestra aleatoria que serán candidatos en cada división ($mtry$); este último parámetro debe ser optimizado mediante un estudio preliminar de la tasa de error OOB.

La figura 5.2 muestra el esquema general de la estrategia basada en MRF. El recuadro interno ilustra el proceso de ajuste y entrenamiento de los algoritmos MRF_{P_p} , ($p = 1, \dots, 7$) así como sus respectivos $RMSE_{MRF_{P_p}}$ que deberán cumplir la condición $RMSE_{MRF_{P_p}} < RMSE_{Tree_{P_p}}$ para la continuidad del proceso. Vale la pena acotar que $RMSE_{Tree_{P_p}}$ corresponde al RMSE producto del ajuste de CART en cada función P_p , ($p = 1, \dots, 7$) tal que si $RMSE_{MRF_{P_p}} > RMSE_{Tree_{P_p}}$ se deberán ajustar nuevamente los parámetros iniciales de CART en la función P_p .

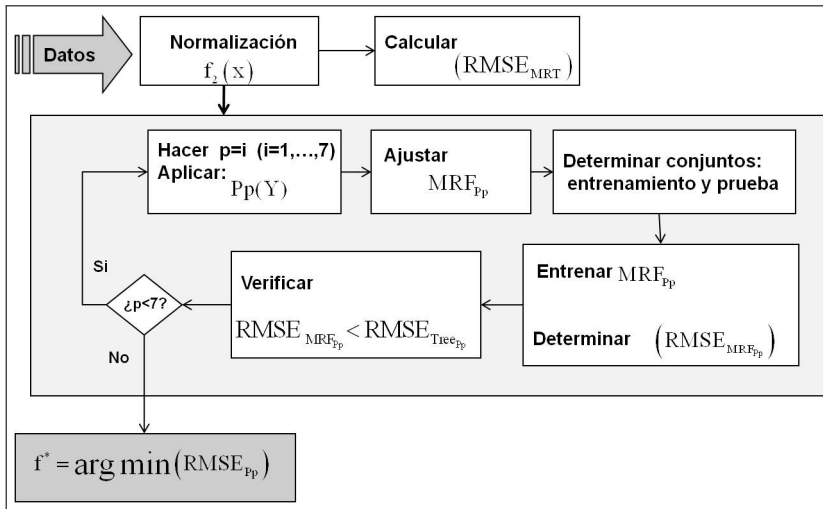


Figura 5.2: Estrategias basadas en MRF

Los pasos a seguir en el esquema general de modelización se detallan a continuación.

Estrategia basada en MRF

Inicialización: Normalizar el conjunto de datos empleando la función de normalización $f_2(x)$ definida en la tabla 4.1.

Paso 1: Ajustar MRT y determinar $RMSE_{MRT}$

Paso 2: Ajustar y entrenar algoritmos MRF_{Pp} siguiendo los siguientes pasos:

- 2.1 Hacer $p = i$ donde $i = 1, 2, \dots, 7$
- 2.2 Aplicar la función Pp al vector de respuestas Y .
- 2.3 Ajustar $Tree_{Pp}$ mediante el conjunto de datos obtenidos en el paso anterior para determinar los parámetros $minsplit$,

$minbucket$ y cp así como su correspondiente $MRSE$, el cual llamaremos $MRSE_{TreeP_p}$.

2.4 Ajustar MRF_{P_p} determinando n_{tree} y $mtry$.

2.5 Dividir aleatoriamente la muestra en conjunto de entrenamiento ($train$) y conjunto de validación o prueba ($test$).

2.6 Entrenar el modelo MRF_{P_p} con el conjunto de entrenamiento y obtener las predicciones de las respuestas para el conjunto de prueba mediante dicho modelo.

2.7 Calcular $RMSE_{MRF_{P_p}}$ de las predicciones obtenidas para el conjunto de prueba. Si $MRSE_{TreeP_p} < RMSE_{MRF_{P_p}}$ volver a 2.3, en caso contrario ir a 2.8

2.8 Si $p < 7$, hacer $p = i + 1$ y repetir proceso, caso contrario ir al paso 3.

Paso 3: Determinar $f^* = argmin(RMSE_{MRF_{P_p}})$, es decir, el algoritmo MRF_{P_p} que proporcione el menor RMSE.

5.3. Estrategias basadas en ANN

Como se comentó a inicios del presente apartado, Su y Chang (2000) propusieron el empleo de las ANN para la modelización de sistemas dinámicos para el caso de una variable respuesta. Nuestro trabajo aporta un método alternativo a dicha propuesta que, como se pudo demostrar, resulta más estable y eficiente estadísticamente. Hemos visto como existen diversas propuestas para resolver el problema en el caso de diseños con múltiples respuestas, entre las cuales citamos nuevamente los trabajos de Chang (2006, 2008), donde las ANN son aplicados nuevamente en la fase de la modelización. Básicamente la propuesta de dichos autores, en esta fase,

corresponde a una simple extensión del caso univariante, el cual se detalló en el capítulo 4 del presente trabajo y que esquematizamos mediante la figura 4.3.

Por todo lo anterior, hemos decidido no repetir los pasos de dicho algoritmo, pero si lo citamos en la presente sección pues en la siguiente, mediante 2 casos de ilustración, usaremos los resultados obtenidos en el trabajo de Chang (2008) y posteriormente los resultados de Chang y Chen (2011) para comparar la estabilidad y eficiencia que nuestra propuesta aporta.

5.4. Casos de ilustración

La presente sección tiene como objetivo ilustrar los pasos para la modelización del vector de respuestas, explicados anteriormente. Se consideran 2 casos de estudio con la finalidad de hacer un análisis numérico exhaustivo. El primer caso se emplea como una prueba piloto para el estudio de cada algoritmo MRF_{Pp} , extendiendo los resultados al segundo caso de estudio para así, mediante un conjunto de datos diferente, verificar las virtudes y deficiencias de la metodología propuesta. Dichos casos corresponden a estudios adoptados por Chang (2008) y Chang y Chen (2011). Ambos casos constan de 3 variables respuestas (y_1, y_2, y_3) obtenidas mediante simulación Monte Carlo y considerando el método de Park y Yum (2003), donde el objetivo es la optimización simultánea de las 3 respuestas bajo las características de calidad definidas en el capítulo 2 como *aproximación a un valor nominal*, *aproximación al valor más pequeño* y *aproximación al valor más grande*: DLB, DNB y DSB respectivamente.

Dividimos la sección en 3 partes. Iniciamos con el ajuste de los parámetros que controlan los algoritmos de modelización para los 2 casos de ilustración, además damos a conocer el lenguaje y las respectivas librerías que fueron empleadas para la programación de las rutinas. En la segunda parte, correspondiente al primer caso de estudio, se presenta una breve descripción del conjunto de datos, seguidamente se determinan las métricas de modelaje para el vector de respuestas y se establece el algoritmo de modelización para posteriormente hacer las comparaciones numéricas entre todos los esquemas estudiados en el presente capítulo y los resultados obtenidos en los trabajos de Chang (2006, 2008). En la tercera parte aplicamos al segundo caso de estudio el algoritmo de modelización que obtuvo los mejores resultados numéricos del caso anterior, con el fin de

contrastarlo con los resultados obtenidos en el trabajo de Chang y Chen (2011).

5.4.1. Parámetros de control

Análogamente al caso univariante, es necesario el ajuste de los parámetros de control de las técnicas empleadas en los algoritmos de modelización. Resaltamos que todos los procedimientos y rutinas empleadas en el diseño de nuestros algoritmos han sido programadas bajo el lenguaje R (versión 2.13.1), donde además ha sido necesario el empleo de algunas librerías, las cuales mencionamos en cada caso. Resaltamos que dichos parámetros son comunes para los dos casos de estudio, con excepción del parámetro *mtry*, que depende propiamente del conjunto de datos.

Algoritmo PRED-MRF

- Ajuste MRT: *minsplit*= 10, *minbucket*= 2, *cp*=0,0001, *xval*= 10, librería *mvpart*, versión 3.1-46 (De'ath, 2011).
- Ajuste CART: *minsplit*= 10, *minbucket*= 2, *cp*=0,0001, *xval*= 10, librería *rpart*, versión 3.1-46 (Therneau y Atkinson, 2010).
- Ajuste MRF: librería *randomForest* versión 4.5-33 (Liaw y Matthew, 2009), $n_{Tree} = 1000$, estimación del RMSE mediante OOB y no se establece criterio de poda entre los árboles.

5.4.2. Caso 5.1

Descripcion del conjunto de datos

Los datos se presentan bajo una matriz ortogonal L_{18} y consta de 6 factores de control $(x_1, x_2, x_3, x_4, x_5, x_6)$, como se muestra en la tabla 5.2. Los datos

se muestran en el anexo B, tabla 12.3. El factor señal posee 3 niveles $M_1 = 10$, $M_2 = 20$ y $M_3 = 30$, y el factor ruido (N) posee 2 niveles N_1 y N_2 . Los valores de especificación del vector de respuestas $\mathbf{Y} = \{y_1, y_2, y_3\}$ con características de calidad DLB, DNB y DSB respectivamente se muestran en la tabla 5.3.

Considerando el enfoque AS y en concordancia con el trabajo presentado por Chang (2008) a efecto de las comparaciones pertinentes, el conjunto de datos es dividido aleatoriamente en dos conjuntos. El conjunto de entrenamiento que consta de 185 observaciones y el conjunto de prueba con 31 observaciones.

Tabla 5.2: Localización de los factores de control en L_{18}

Nº	x_1	x_2	x_3	x_4	x_5	x_6
1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	3	3	3	3	3	3
4	1	1	2	2	3	3
5	2	2	3	3	1	1
6	3	3	1	1	2	2
7	1	2	1	3	2	3
8	2	3	2	1	3	1
9	3	1	3	2	1	2
10	1	3	3	2	2	1
11	2	1	1	3	3	2
12	3	2	2	1	1	3
13	1	2	3	1	3	2
14	2	3	1	2	1	3
15	3	1	2	3	2	1
16	1	3	2	3	1	2
17	2	1	3	1	2	3
18	3	2	1	2	3	1

Tabla 5.3: Especificación de las respuestas. Caso 5.1

Respuestas	y_1	y_2		y_3
Límites	y_{jk}^{\min}	y_{jk}^{\min}	y_{jk}^{\max}	y_{jk}^{\max}
M_1	55	7	13	3
M_2	110	14	26	6
M_3	165	21	39	9

Análisis de las métricas de modelaje: resultados numéricos

La figura 5.3 muestra los valores de RMSE alcanzados por el ajuste inicial de MRT y los obtenidos por los MRF en las distintas métricas, es decir $RMSE_{MRT}$ y $RMSE_{MRF_{P_p}}$, ($p = 1, \dots, 7$) establecidos en el paso 2 del algoritmo de modelización. Se puede notar, con gran sorpresa, que P_7 resulta totalmente ineficiente en la obtención del RMSE con valores incluso superiores al arrojado por MRT . Así pues, dicha función es excluida del abanico de posibles funciones de modelización.

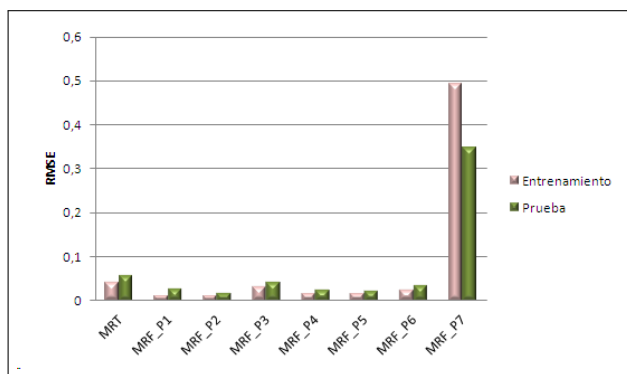


Figura 5.3: Comparación del RMSE en las métricas de modelaje

La tabla 5.4 refleja los valores de RMSE obtenidos hasta ahora, donde el objetivo es la determinación del algoritmo que proporcione el menor RMSE, es decir $f^* = \operatorname{argmin}(RMSE_{MRF_{P_p}})$. Para ello, en dicha tabla hemos definido una medida de *progreso* $\mathbf{P}\%$ de los valores alcanzados en cada algoritmo MRF_{P_p} frente al MRT ajustado inicialmente, observando tanto el conjunto de entrenamiento como el conjunto de prueba. Dicha medida la expresamos en porcentaje (%). Vemos como el efecto cuadrático de los datos nos conduce a valores menores de RMSE en la funciones P_2, P_4 y P_5 y en consecuencia un mejor rendimiento del algoritmo de modelización, 70,60 %, 63,19 % y 63,43 % en el conjunto de entrenamiento

y 42,81 % 32,53 % y 35,27 % en el conjunto de prueba, respectivamente. En contraste con ello, podemos ver como la función lineal P_1 alcanza un rendimiento cercano a P_2 , 69,68 % en el conjunto de entrenamiento, pero en el conjunto de prueba los rendimientos son incluso inferiores a P_4 y P_5 lo que se traduce en sobreajuste del modelo MRF_{P_1} . Ahora bien, el objetivo en la fase de modelización es obtener una función objetivo lo más robusta posible, por tanto el tema del sobreajuste debe ser minimizado. Entonces se elige a P_2 como la función de globalización para el diseño del MRF, ya que presenta las menores discrepancias entre los conjuntos de entrenamiento y prueba. En lo sucesivo denotaremos a dicho algoritmo de modelización como MRF_{P_2} .

Tabla 5.4: Progreso en los valores de RMSE de los algoritmos MRF_{P_p} frente a MRT. Caso 5.1

Función	RMSE		P %	
	Entrenamiento	Prueba	Entrenamiento	Prueba
MRT	0.0432	0.0584	–	–
MRF_{P_1}	0.0131	0.0264	69.68	28.77
MRF_{P_2}	0.0127	0.0182	70,60	42,81
MRF_{P_3}	0.0327	0.0429	24.31	0.51
MRF_{P_4}	0.0159	0.0242	63.19	32.53
MRF_{P_5}	0.0158	0.0226	63.43	35.27
MRF_{P_6}	0.0236	0.0347	45.37	14.55
MRF_{P_7}	0.492	0.348	-1038.89	-521.23

Implementación Random Forest multivariante

Luego de determinar la función de normalización de los datos y la métrica de modelaje del vector de respuestas Y , se estudia el parámetro $mtry$ mediante la tasa de error OOB como se ilustra en la figura 5.4, tal que se fija $mtry = 4$ para su incorporación en el establecimiento del RFM.

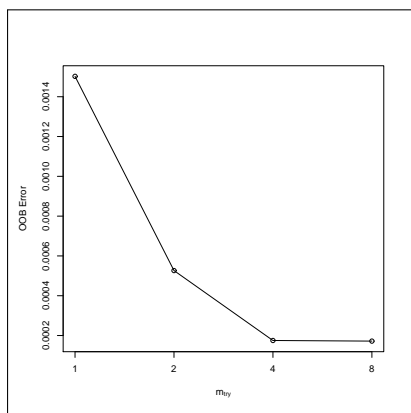


Figura 5.4: Ajuste del parámetro m_{try} mediante la tasa de error OOB. Caso 5.1

Finalmente, nuestro algoritmo MRF_{P_2} para el presente caso de estudio se establece como sigue.

Inicialización: Normalizar el conjunto de datos mediante $f_2(x)$ (tabla 4.1).

Paso 1: Ajustar $Tree_{P_2}$ considerando $minsplit = 10$, $minbucket = 2$, $cp = 0,0001$ y determinar $MRSE_{MRT}$.

Paso 2: Ajustar y entrenar algoritmo MRF_{P_2} siguiendo los siguientes pasos:

- 2.1** Aplicar la función P_2 al vector de respuestas Y .
- 2.2** Ajustar MRF_{P_2} considerando $n_{tree} = 1000$ y $mtry = 4$.
- 2.3** Dividir aleatoriamente la muestra en conjunto de entrenamiento y prueba.
- 2.4** Entrenar el modelo MRF_{P_2} con el conjunto de entrenamiento y obtener las predicciones de las respuestas para el conjunto de prueba.

- 2.5** Calcular $RMSE_{MRF_{P_2}}$ de las predicciones obtenidas para el conjunto de prueba. Si $RMSE_{Tree} < RMSE_{MRF_{P_2}}$ volver a 2.2, en caso contrario ir al paso 3.

Paso 3: Determinar función objetivo f^* .

Comparación del modelo predictivo MRF con ANN

Como se ha podido ver a inicios del presente trabajo, existen diversas propuestas en torno al problema del diseño de parámetros en casos de múltiple respuestas. Entre las más recientes encontramos los trabajos de Chang (2006, 2008). Ambos trabajos tienen en común el establecimiento de las ANN como fase inicial en la modelización de la respuesta. Así pues, considerando que nuestra propuesta representa una alternativa a dichos trabajos, por ser los más actuales en la bibliografía, hemos comparado nuestros resultados numéricamente con los resultados presentados por dichos autores en sus trabajos.

A tal fin se presenta la figura 5.5 donde nos ha parecido pertinente incorporar los resultados obtenidos por los diferentes $RMSE_{MRF_{P_p}}$, a excepción de $RMSE_{MRF_{P_7}}$ que fue excluido dado los resultados discutidos en el punto anterior; todo ello en contraste con los resultados obtenidos por Chang (2006) y Chang (2008). En dicha figura hemos denotado como ANN7 – 13 – 3 y ANN8 – 12 – 3 a los resultados de RMSE obtenidos por Chang (2006) y Chang (2008) respectivamente, donde la numeración en ambos casos corresponde a las arquitecturas consideradas por dichos autores como la mejor, a pesar de tratarse del mismo conjunto de datos. A simple vista se observa que nuestros resultados resultan mejores (menores valores de RMSE), incluso considerando cualquier función P_p tanto en el conjunto de entrenamiento como en el conjunto de prueba, lo que hace a

nuestro método robusto estadísticamente.

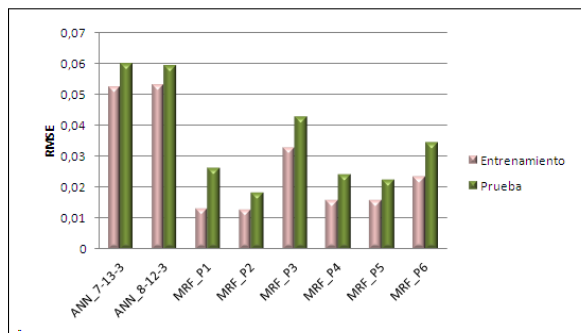


Figura 5.5: Comparación de los $RMSE_{MRF_{P_j}}$ con otras propuestas

Para finalizar el presente caso de ilustración se presenta la tabla 5.5, donde se comparan los resultados obtenidos por $RMSE_{MRF_{P_2}}$ con los trabajos realizados por Chang (2006, 2008) considerando nuevamente los conjuntos de entrenamiento y prueba, así como el número de iteraciones necesarias por cada algoritmo. Definimos en este caso la medida de *progreso* $\mathbf{P} \%$ de los valores alcanzados por MRF_{P_2} frente a los RMSE alcanzados por Chang (2006) y Chang (2008), respectivamente. Se observa, en contraste con el trabajo más reciente, que con el empleo de MRF_{P_2} se obtiene un progreso de 76 % y 69 % en los conjuntos de entrenamiento y prueba respectivamente con tal solo 1000 iteraciones, lo que sitúa a MRF_{P_2} por encima de tales trabajos en términos de RMSE y por ende en robustez estadística que es el objeto de la fase de modelización.

Tabla 5.5: Progreso de MRF_{P_2} frente a otras metodologías. Caso 5.1

Propuesta	Iteraciones	RMSE		$\mathbf{P} \%$	
		Entrenamiento	Prueba	Entrenamiento	Prueba
Chang (2006)	30000	0.0524	0.0601	75.76	69.72
Chang (2008)	4500	0.0573	0.0595	76.17	69.41
MRF_{P_2}	1000	0.0127	0.0182		

5.4.3. Caso 5.2

Descripción del conjunto de datos

Los datos se presentan bajo una matriz ortogonal L_{18} , análogamente a la presentada en la tabla 5.2, pero en este caso el vector de factores de control se define como $X = \{A, B, C, D, E, F\}$, el factor señal definido con los niveles $M_1 = 0,1$, $M_2 = 0,2$ y $M_3 = 0,3$ y el experimento es conducido con 2 repeticiones. Los datos se presentan en el anexo B, tabla 12.4. Los valores de especificación del vector de respuestas $\mathbf{Y} = \{y_1, y_2, y_3\}$ con características de calidad DLB, DNB y DSB, respectivamente, se muestran en la tabla 5.6. A fines comparativos con los resultados de Chang y Chen (2011), el conjunto de entrenamiento y prueba se conforma aleatoriamente con 92 y 16 observaciones respectivamente.

Tabla 5.6: Especificación de las respuestas. Caso de ilustración 5.2

Respuestas	y_1	y_2		y_3
Límites	y_{jk}^{\min}	y_{jk}^{\min}	y_{jk}^{\max}	y_{jk}^{\max}
M_1	4.8	0.6	1.4	28
M_2	9.6	1.2	2.8	56
M_3	14.4	1.6	4.2	84

Implementación Random Forest multivariante

Como se dijo al inicio de la presente sección, este caso de estudio hereda los resultados obtenidos en el caso 1, es decir, realizamos los análisis numéricos considerando la función de normalización f_2 (tabla 4.1) y la métrica P_2 de la tabla 5.3. Pero considerando que se trata de una base de datos diferente, es necesario el ajuste del parámetro m_{try} , el cual se muestra en la figura 5.6 y mediante la cual se fija $m_{try} = 2$.

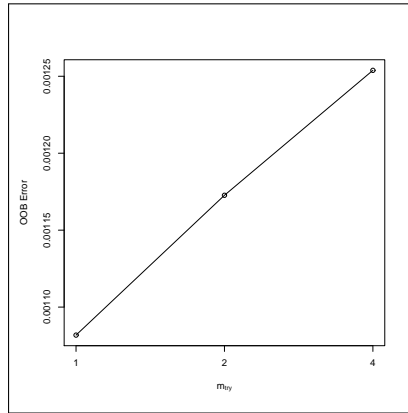


Figura 5.6: Ajuste del parámetro m_{try} mediante la tasa de error OOB.
Caso 5.2

Finalmente, el esquema de modelización para el presente caso de estudio queda estructurado como sigue.

Inicialización: Normalizar el conjunto de datos empleando la función de normalización $f_2(x)$ definida en la tabla 4.1.

Paso 1: Ajustar MRT haciendo $minsplit = 10$, $minbucket = 2$, $cp = 0,0001$ y determinar $MRSE_{MRT}$.

Paso 2: Ajustar y entrenar algoritmo MRF_{P_2} siguiendo los siguientes pasos:

- 2.1** Aplicar la función P_2 al vector de respuestas Y .
- 2.2** Ajustar MRF_{P_2} haciendo $n_{tree} = 1000$ y $m_{try} = 2$ (figura 5.6).
- 2.3** Dividir aleatoriamente la muestra en conjunto de entrenamiento y conjunto de prueba.
- 2.4** Entrenar el modelo MRF_{P_2} con el conjunto de entrenamiento y obtener las predicciones de las respuestas para el conjunto de prueba.

- 2.5** Calcular $RMSE_{MRF_{P_2}}$ de las predicciones obtenidas para el conjunto de prueba. Si $RMSE_{MRT} < RMSE_{MRF_{P_2}}$ volver a 2.2, en caso contrario ir al paso 3.

Paso 3: Determinar función objetivo f^* .

Comparación del modelo predictivo MRF con ANN

La figura 5.7 compara los RMSE obtenidos de nuestro algoritmo de modelización MRF_{P_2} con el mejor resultado presentado por Chang y Chen (2011), el cual es basado en ANN. A simple vista se puede notar que, si bien es cierto que la diferencia entre los RMSE para el conjunto de entrenamiento no es muy grande, también es cierto que, en el conjunto de prueba dichas diferencias si son notorias. Además, MRF_{P_2} presenta menos discrepancias entre los RMSE de los conjuntos de entrenamiento y prueba en comparación con los resultados de Chang y Chen (2011) para dichos conjuntos, lo que se traduce no solo en buen rendimiento sino en robustez estadística.

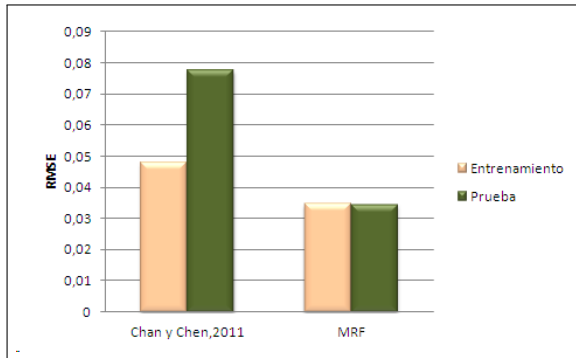


Figura 5.7: Comparación del RMSE de MRF_{P_2} con los trabajos realizados por Chang y Chen (2011)

A fin de cuantificar lo observado anteriormente se presenta la tabla 5.7

donde nuevamente definimos la medida $\mathbf{P} \%$ como el *progreso* que implica el empleo de MRF_{P2} frente a la propuesta de Chang y Chen (2011). Note como que MRF_{P2} conlleva un progreso del 55% en el conjunto de prueba, lo que verifica su robustez estadística frente a Chang y Chen (2011); además, resaltamos que nuestros rendimientos fueron obtenidos con solamente 1000 iteraciones en contraste con las 10000 iteraciones necesarias en las ANN.

Tabla 5.7: Comparación de los RMSE en consideración al número de iteraciones.Caso 5.2

Propuesta	Iteraciones	RMSE		$\mathbf{P} \%$	
		Entrenamiento	Prueba	Entrenamiento	Prueba
Chang y Chen (2011)	10000	0.0479	0.0776	27.56	55.41
MRF_{P2}	1000	0.0347	0.0346		

Capítulo 6

Estrategias definitivas de modelización

El presente capítulo desarrolla las conclusiones de la fase inicial de nuestra propuesta metodológica, la cual tiene como objetivo la modelización de la respuesta y establecimiento de la función objetivo en virtud de no poseer un modelo a priori. Todo ello considerando problemas de tipo univariante y posteriormente generalizado al caso multivariante.

En el capítulo 3 mostramos las bases teóricas que dieron lugar al diseño de las estrategias con el fin de evitar vacíos conceptuales. En el capítulo 4 propusimos 3 funciones de normalización con el fin de minimizar la variabilidad presente en el conjunto de datos y unificar escalas con las ANN para comparaciones posteriores. Se seleccionó la mejor función de normalización en virtud a dos criterios: la dispersión de los datos y el rendimiento general de los algoritmos tal que los métodos predictivos que la incorporen presenten mayor robustez. Todo ello para dar respuesta al objetivo específico 1. Seguidamente, considerando el caso univariante,

hemos diseñado diversas estrategias de modelización que combinan técnicas de minerías de datos, las hemos comparado numéricamente y estudiamos sus propiedades teóricas en respuesta a los objetivos específicos 2 y 3. Como una generalización de los resultados anteriores e incluyendo las métricas acordadas se desarrolló el capítulo 5, donde se diseñaron las estrategias de modelización para casos multivariantes y comparamos nuestros resultados con estrategias basadas en ANN, todo esto en respuesta a los objetivos específicos 4 y 5.

A continuación se presenta el resumen de los aspectos más destacados, tanto en el caso univariante como en el multivariante, así como el establecimiento del esquema de modelización definitivo que permitirá el paso a la siguiente fase objeto del presente trabajo, que es la optimización en base a los modelos establecidos.

6.1. Metodología PRED-RF

A lo largo del capítulo 3 se presentaron las virtudes y deficiencias de CART y ANN como dos herramientas poderosas en procesos de modelización. En el caso de CART como estimadores débiles, mostramos que una forma de reducir su inestabilidad es su combinación con algunas técnicas de remuestreo (Validación cruzada, Bootstrap y Bagging). Por tanto, propusimos estrategias de modelización que combinan dichas técnicas de remuestreo con los árboles de regresión pero percibimos, numéricamente, que dichas técnicas a su vez encierran algunas limitaciones que perfectamente son transmitidas a los árboles de regresión, y por ende su mejora en términos de variabilidad se ve limitada. Ante tal situación, diseñamos un esquema de modelización basado en RF, en consideración a que ataca la variabilidad presente en los árboles de regresión mediante muestras Bootstrap y

Bagging. Propusimos además una fase de preparación previa del conjunto de datos mediante funciones de normalización con el fin de mejorar la robustez de las estrategias diseñadas. Finalmente y con fines comparativos, agregamos a nuestro abanico de propuestas la estrategia presentada en el trabajo de Su y Chang (2000) la cual está basada en el empleo de ANN.

Proponemos el algoritmo PRED-RF como estrategia de modelización basada en CART y RF, cuyo objetivo es determinar de forma eficiente y robusta la función objetivo. Acotamos que para el diseño de dicho esquema partimos de una matriz ortogonal L_{18} pero se puede extender perfectamente a otros tipos de matrices. Los factores pueden ser cualitativos, cuantitativos (discretas o continuas) o incluso una mezcla entre las mismas; lo que le permite a PRED-RF ser flexible a casos donde se consideren factores de señal o ruido. No suponemos independencia entre los factores ni relaciones lineales con la variable respuesta, por tanto puede ser adoptado en casos de correlaciones y no linealidad.

Mediante 2 conjuntos de datos reales se realizaron los estudios y contrastes numéricos pertinentes para la determinación del esquema de modelaje y dar continuidad a la siguiente fase de nuestra propuesta metodológica. Iniciamos con el estudio de las funciones de normalización donde, considerando que el objetivo es reducir la variabilidad propia del conjunto de datos, se realizaron análisis de varianzas (ANOVA), los cuales nos llevaron a elegir la función que se presenta a continuación y que fue integrada en los esquemas de modelización basados en CART y RF para los análisis posteriores.

$$f_2(x) = \begin{cases} 0 & x = 0 \\ \frac{1}{1+\exp(-x')} & \text{en otros casos} \end{cases} \quad \text{donde } x' = \frac{x - \min}{\min - \max}$$

Numéricamente se pudo observar como PRED-RF presentó los mejores ajustes en términos de RMSE como estadístico de prueba. Se evidenció que PRED-RF alcanza buenos resultados en las distintas opciones en cuanto a número de iteraciones realizadas, lo que se traduce en estabilidad estadística. Por su parte, las ANN no sólo necesitan gran número de iteraciones para estabilizar sus resultados sino, que presentan problemas de sobreajuste cuando se comparan los conjuntos de entrenamiento y prueba. Por todo lo anteriormente dicho y considerando que la siguiente fase se nutre de los resultados aquí obtenidos presentamos, mediante el siguiente flujograma, nuestra propuesta definitiva de modelización para la determinación de la función objetivo a ser optimizada en el siguiente apartado.

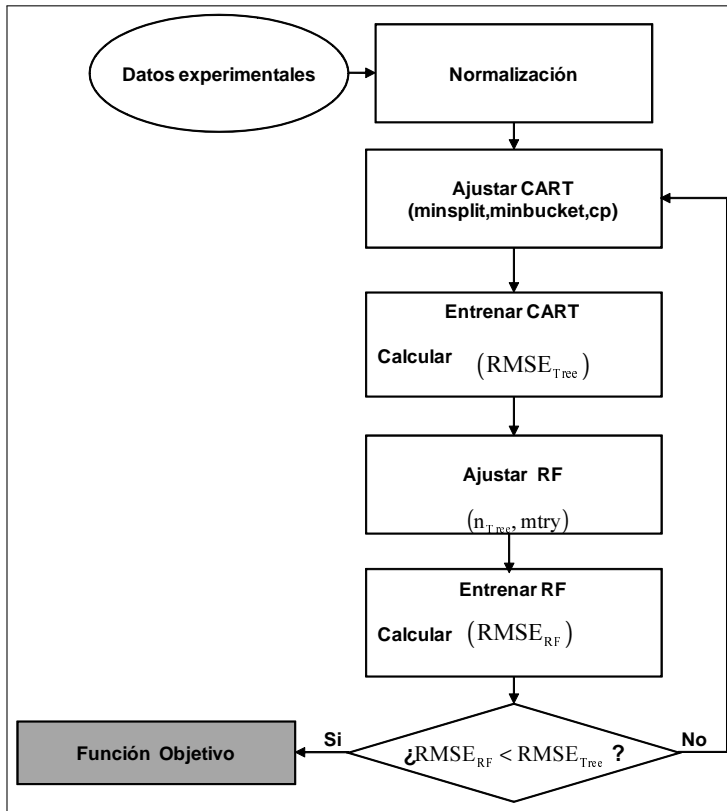


Figura 6.1: Metodología PRED-RF

6.2. Metodología MRF

Los árboles de regresión multivariantes (MRT) constituyen una extensión del algoritmo CART a casos de múltiples variables respuestas; por tanto, hereda sus buenas propiedades pero también su condición de sensibilidad a la determinación de los conjuntos de entrenamiento y prueba. Entonces, ante el uso de MRT nos arriesgamos a una situación de variabilidad no permisible en nuestro objetivo de modelización. En vista a los buenos resultados obtenidos en nuestra propuesta al caso univariante, decidimos diseñar un esquema de modelización para casos de múltiples respuestas basado en CART y RF.

Así pues, nuestra propuesta de modelización para casos multivariantes consiste en una estrategia que hemos definido *Multivariante Random Forest* (MRF), cuyo diseño involucra fases previas de normalización, métricas de modelaje y el ajuste de MRT como medida límite en términos de RMSE. Los análisis correspondientes a nuestra propuesta se llevaron a cabo mediante 2 casos de ilustración diferentes. Usamos el primer caso como prueba piloto de los esquemas de modelización y mediante el segundo caso de ilustración se verificaron los resultados obtenidos. Todo ello nos permitió además comparar nuestros resultados con los trabajos presentados por Chang (2006, 2008) y Chang y Chen (2011) respectivamente.

Acotamos que la fase de normalización se realizó empleando la función f_2 definida en la tabla 4.1 y que obtuvo los mejores resultados numéricos en el caso univariante; pero a efectos de futuros estudios se recomienda el estudio preliminar de las diferentes funciones de modelización tal como se explica en el capítulo 4 del presente trabajo.

Para la determinación de la métrica de modelaje del vector de respuestas Y se diseñaron 7 expresiones, considerando las posibles correlaciones entre las variables respuestas, así como sus posibles efectos cuadráticos. Dichas expresiones fueron contrastadas entre sí usando como estadístico de prueba los valores de RMSE en cada caso. Vale la pena resaltar que, en vista a la robustez de RF en el caso univariante decidimos entrenar nuestro MRF con 1000 iteraciones, donde la expresión que presentó los mejores resultados consiste en la media geométrica de las r variables del vector Y en cada m -ésima respuesta y que se muestra a continuación.

$$P_2(Y) = \left(\prod_{i=1}^r y_r \right)^{\frac{1}{r}}$$

Numéricamente se establecieron comparaciones con las estrategias basadas en ANN. Donde, análogamente al caso univariante, pudimos verificar las virtudes de nuestro esquema de modelización. Nuestra propuesta logró ajustar un RMSE 5 veces menor en 1000 iteraciones en comparación con las ANN, quienes necesitan 4500 iteraciones para alcanzar su valor mínimo de RMSE. Todo ello se presentó en el caso piloto de ilustración. En cuanto al segundo caso de ilustración, los resultados no fueron menos relevantes, nuestro MRF no solo logró ajustar menores valores de RMSE sino que ahora, mediante un conjunto de datos diferente, verificamos que la diferencia entre los conjuntos de entrenamiento y prueba sólo se sitúa en el 0,01 %, a diferencia de las ANN cuya diferencia alcanza el 2,97 %. Por otra parte, es de notar que para tales resultados nuestro MRF sólo necesitó 1000 iteraciones, en contraposición con las 10000 empleadas por las ANN.

Finalmente y considerando que establecer el esquema de modelización es la primera etapa en el establecimiento de nuestra propuesta metodológica,

presentamos mediante la figura 6.2 el esquema definitivo para la determinación de la función objetivo, el cual será el punto de partida en la fase de optimización a tratar en la parte III.

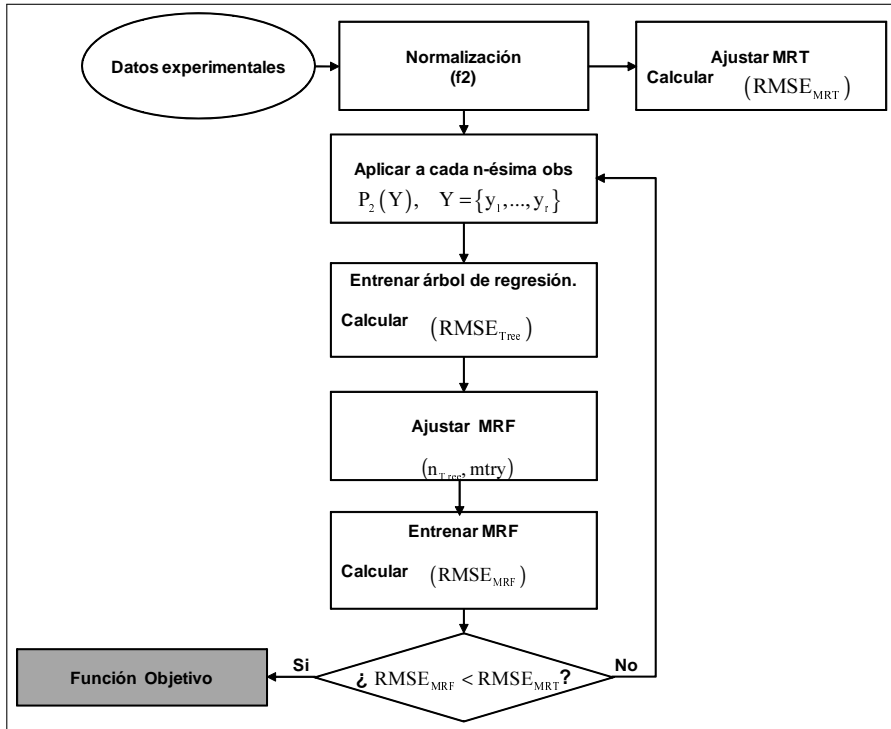


Figura 6.2: Metodología MRF

Parte III

Estrategias para la optimización de sistemas

Capítulo 7

Introducción

Los métodos heurísticos son procedimientos que tratan de descubrir una buena solución factible, que tal vez no sea la óptima, para un problema específico en consideración. No puede darse una garantía acerca de la calidad de la solución que se obtiene, pero cuando están bien diseñados pueden proporcionar soluciones bien cercanas al óptimo real. Éstos procedimientos se basan en ideas relativamente simples de sentido común acerca de la forma en que se debe buscar una buena solución, pero son de naturaleza *ad hoc*, es decir, por lo general cada método se diseña para abordar un tipo específico de problema en vez de una variedad de aplicaciones. Nacen entonces los métodos *metaheurísticos* proporcionando estructuras generales con criterios estratégicos para desarrollar un método heurístico específico que se ajuste a un tipo particular de problema (Hillier y Lieberman, 2006).

Así pues, la metaheurística es un método de solución general que organiza la interacción entre los procedimientos de mejora local y las estrategias de más alto nivel, creando procesos capaces de escapar de un óptimo

local y realizar búsquedas rigurosas en el campo de acción. Entre las metaheurísticas mayormente reseñadas en procesos de optimización se encuentran Simulated Annealing (AS) y los Algoritmos Genéticos (GA, por sus siglas en inglés) presentándose éstos últimos, en algunos trabajos, como *económicos* en tiempos de ejecución ya que inician el proceso de búsqueda con una población de posibles soluciones en lugar de una posible solución; es necesario resaltar que la eficiencia de cada metaheurística está ligada al tipo de problema y su complejidad (Degertekin, 2007; Tsenov, 2006). Sin embargo, consideramos que los GA poseen buenas propiedades a efectos del presente trabajo, ya que su búsqueda esta basada en la exploración simultánea de hiperplanos, muestreados por numerosos puntos en la población, lo que es conocido como la propiedad de *paralelismo implícito* y que presenta a los GA como métodos de optimización robusta (Lucasius y Kateman, 1993).

Nuestra propuesta consiste en un híbrido entre las estrategias de modelización resultantes de la fase de modelización anterior y los GA como metaheurística. Por tanto, éste capítulo se inicia con los fundamentos teóricos más relevantes de los GA y finaliza con una breve reseña de las funciones de deseabilidad, como nuestra función de evaluación para el desarrollo del GA en el caso multivariante.

7.1. Algoritmos Genéticos

Los algoritmos genéticos (GA) son métodos adaptativos inspirados con la *teoría biológica de la evolución* formulada por Darwin a mediados del siglo XIX. En la naturaleza, los individuos de una población permanecen en constante competición por recursos como agua, refugio y comida. Los indi-

viduos con mayor éxito en tal lucha tienen más probabilidades de sobrevivir y de tener una descendencia mayor. Al contrario, individuos peor adaptados tienen un número menor de descendientes, o incluso ninguno. Todo ello implica que los genes de los individuos mejor adaptados se propagarán a través de las generaciones. La combinación de características buenas a través de los ancestros puede dar lugar a descendencias mejor adaptadas que los padres. De esta manera, la especie evoluciona adaptándose cada vez más a su entorno a través de las generaciones.

Tales ideas se transfieren a los problemas de optimización de forma bastante natural. Las soluciones factibles de un problema específico corresponden a los miembros de una especie particular, donde la aptitud de cada miembro se mide por el valor de la función objetivo. La *población* actual en cada iteración (generación) consiste en un conjunto de soluciones de prueba y se entiende como los miembros vivos de la especie. Algunos de los miembros más jóvenes de la población (en especial los miembros más aptos) sobreviven a la adultez y se convierten en *padres* (aparejados de forma aleatoria) que tendrán *hijos* (nuevas soluciones de prueba) que tienen algunas de las características (genes) de los padres. Como los miembros más aptos de la población tienen una mayor probabilidad de convertirse en padres que los otros, el GA tiende a crear *poblaciones mejoradas* a medida que avanzan las generaciones. De vez en cuando ocurren *mutaciones*, de modo que las nuevas generaciones pueden adquirir características que no poseen los padres. Éste fenómeno ayuda a los GA a explorar una parte de la región factible, quizás mejor que la considerada con anterioridad. Finalmente, la supervivencia del más apto tiende a conducir al GA hacia una solución de prueba (la mejor de todas las consideradas) que al menos es cercana al óptimo.

Aunque la analogía con el proceso de la evolución biológica define la esencia de cualquier GA, no es necesario adherirse rígidamente a dicha analogía. Por ejemplo, algunos GA permiten que la misma solución de prueba sea un padre en forma repetida a través de múltiples generaciones. De modo que la analogía debe ser considerada como punto de partida para definir los detalles del GA que se ajuste mejor al problema bajo consideración (Hillier y Lieberman, 2006).

7.1.1. Componentes de los Algoritmos Genéticos

Nuestra propuesta de optimización se orienta en un GA simple, por tal razón se presenta el respectivo esquema mediante la figura 7.1 y en las siguientes secciones se hará una descripción breve de cada fase, haciendo especial énfasis en los *operadores genéticos*: evaluación, selección, cruce y mutación; representados en el recuadro interno de dicha figura.

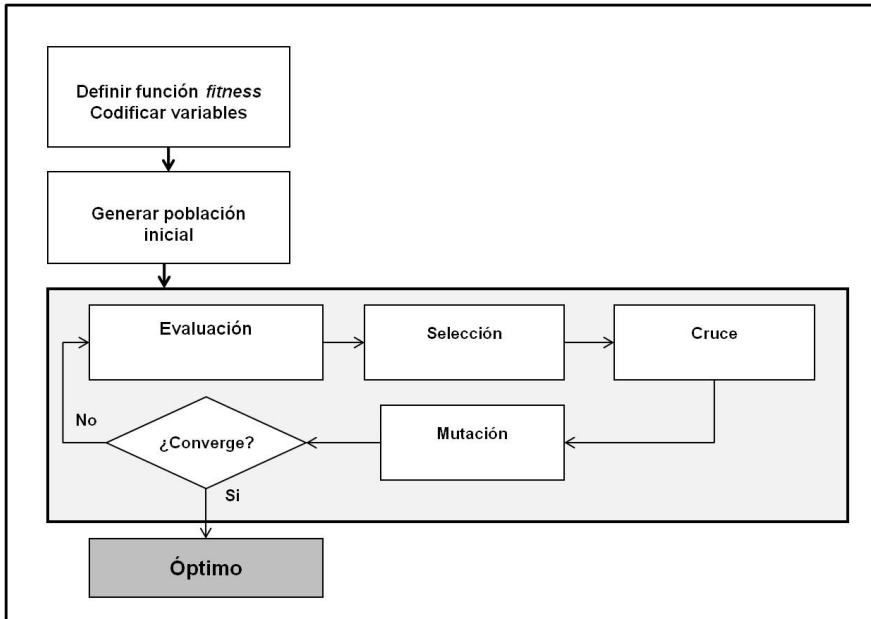


Figura 7.1: Esquema general de un algoritmo genético simple

Función de evaluación y codificación de variables

Para el correcto funcionamiento de un GA se debe poseer un método (función) que evalúe el comportamiento de cada individuo de la población. Por lo tanto, para cada tipo de problema que se desee resolver deberá derivarse una función de evaluación, lo cual comúnmente es llamada como *fitness*. Dicha función será la responsable de establecer la medida de coste del ajuste de cada individuo, por tanto marcará la convergencia o no del GA y las aplicaciones de los operadores genéticos (selección, cruce y mutación).

Otro punto fundamental es la codificación de las variables, es decir, la determinación de la estructura de los cromosomas. Tradicionalmente, dichas codificaciones se realizaban de forma binaria, es decir, se asigna un determinado número de *bits* a cada parámetro y se realiza una

discretización de la variable representada por cada gen. Lo anterior se ilustra en la expresión (7.1) donde el cromosoma esta representado por g_i genes ($i = 1, \dots, n$) que corresponden a las n variables bajo estudio. En este caso cada variable (gen) ha sido representado con 10 bits

$$cromosoma = [\underbrace{1111001001}_{g_1} \underbrace{0011011111}_{g_2} \dots \underbrace{0000101001}_{g_n}] \quad (7.1)$$

Sin embargo también pueden existir representaciones que codifiquen directamente cada variable con un valor real, valor entero o puntos flotantes (Haupt y Haupt, 2004), tal que el cromosoma no conste de una larga colección de números (bits) tendiendo esto a causar lentitud e imprecisión en el desarrollo del GA (Michalewicz, 1996). Así por ejemplo, la expresión (7.1) se puede simplificar mediante (7.2) donde cada gen x_i representa directamente la i -ésima variable de interés y su valor estará condicionado al dominio de permanencia.

$$cromosoma = [\underbrace{x_1}_{g_1} \underbrace{x_2}_{g_2} \dots \underbrace{x_n}_{g_n}] \quad (7.2)$$

Entonces, si llamamos a f la función de evaluación y tomamos el cromosoma de (7.2), la medida de coste del individuo quedara denotado por

$$coste = f(cromosoma) = f(x_1, x_2, \dots, x_n)$$

Vale la pena acotar que a (7.2) se le define como *cromosoma continuo* y a efectos del presente trabajo no solo se adopta éste tipo notación, sino que a partir de ahora enfocamos nuestro fundamento teórico a los llamados *algoritmos genéticos continuos*.

Población inicial

La población inicial quedará representada por una matriz donde cada fila representa un cromosoma. Cada cromosoma contendrá los n genes que representan las n variables (parámetros del problema), éstos genes son determinados de forma aleatoria pero, obviamente, dentro del rango establecido por cada parámetro. Por ejemplo, si el problema bajo estudio consta de n parámetros, representados por el vector $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, entonces la matriz para la población inicial con r cromosomas tendrá la forma:

$$\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ x_{r1} & x_{r2} & \dots & x_{rn} \end{pmatrix}$$

Tamaño de la población inicial y condición de parada

El tamaño de la población inicial sigue siendo punto de discusión entre los usuarios de ésta metaheurística. Parece intuitivo que las poblaciones pequeñas corren el riesgo de no cubrir adecuadamente el espacio de búsqueda, mientras que trabajar con poblaciones excesivamente grandes puede acarrear problemas relacionados con gastos innecesarios de recursos computacionales y convergencia prematura en algunos casos. Por tanto, el tamaño idóneo de la población de inicio sigue quedando a decisión del usuario en consideración con el tipo de problema a resolver y los recursos computacionales con los que se cuenta (Haupt y Haupt, 2004; Lucasius y Kateman, 1993).

En cuanto a la condición de parada del GA, lo más habitual es que la

condición de terminación sea la convergencia a un valor óptimo (máximo o mínimo, dependiendo del tipo de problema) o un número prefijado de generaciones. En el contexto de los GA, se habla de la convergencia cuando aproximadamente el 95 % de los genes poseen el mismo valor tras la evaluación, cuando se habla de codificaciones binarias, o valores dentro de un rango de respuestas, en caso de codificaciones de tipo continuo. (Haupt y Haupt, 2004; Bandyopadhyay y Sankar, 2007)

Operadores genéticos

Describiremos ahora los operadores genéticos presentes en el proceso recursivo de un GA simple: selección, cruce y mutación; los cuales se han ilustrado en el recuadro interno de la figura 7.1. Dicho proceso dependerá de la convergencia o no en la evaluación de los cromosomas. Esto es, si la evaluación de la población en curso no converge se llevan a cabo los operadores genéticos dando lugar a nuevas y nuevas generaciones por cada iteración hasta que el GA encuentre el punto de convergencia.

Selección

Los algoritmos de selección serán los encargados de determinar qué individuos podrán reproducirse y pasar a la siguiente generación y cuáles no. Puesto que se trata de imitar el proceso de selección natural, se debe otorgar un mayor número de posibilidades a los individuos más aptos, por tanto la selección de un individuo dependerá de su valor de ajuste determinado por la función de evaluación. Éste operador, si no se define adecuadamente, puede causar la convergencia prematura del GA ya que puede crear poblaciones muy homogéneas en pocas iteraciones. Entonces, se recomienda establecer el criterio de selección tal que no se eliminen por completo los individuos poco aptos o medianamente aptos (Haupt y

Haupt, 2004).

Como se dijo anteriormente, no es necesario adherirse rígidamente a las ideas de la evolución natural. Así por ejemplo, se puede optar por selecciones aleatorias o elitistas que permiten mantener al individuo más apto sin alteraciones genéticas en varias generaciones o en caso contrario sirva de padre en varias generaciones. No obstante, una opción bastante común es seleccionar al primer individuo de forma aleatoria y el segundo por el método de torneo. El método del torneo consiste básicamente en realizar la selección en base a comparaciones directas entre individuos, pero existen dos versiones de éste método: la determinista y la probabilística. En la versión determinista se selecciona al azar un número p de individuos y de esos se selecciona el más apto para pasarlo a la siguiente generación; ese proceso se repite reiteradas veces hasta completar el número de individuos a seleccionar. La versión probabilística sólo se diferencia de la versión anterior en que en lugar de elegir el mejor en el grupo de p individuos, se genera un número aleatorio en el intervalo $[0, 1]$ y entonces si es mayor que un número t fijado previamente, se elige el más apto; en caso contrario se toma el menos apto.

Cruce

Una vez seleccionados los individuos, éstos son combinados para producir la descendencia que formará parte de la siguiente generación. La idea de los cruces se basa en que, si se toman dos individuos correctamente con buenos valores de adaptación y se obtiene una descendencia que comparta genes de ambos, existe la posibilidad de que los genes heredados sean precisamente los causantes de la bondad de ajuste de los padres. Existen variedad de métodos de cruce, pero los más reseñados en la literatura consultada son el cruce de 1 y 2 puntos, los cuales detallamos brevemente a continuación.

Cruce de 1 punto: Es la forma más sencilla de cruce, en donde 2 individuos seleccionados (digamos madre y padre) se cortan en un punto elegido de forma aleatoria para generar dos segmentos y posteriormente se intercambian los segmentos como lo ilustra en la figura 7.2. En tal figura se puede apreciar la formación de dos hijos, h_1 y h_2 , mediante el intercambio genético a partir del punto x_{13} y x_{33} del padre y la madre respectivamente. Vale la pena acotar que este tipo de cruce es el adoptado en nuestra propuesta de optimización.

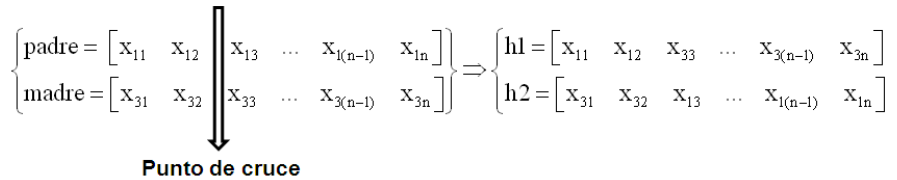


Figura 7.2: Cruce de 1 punto

Cruce de 2 puntos: Se trata de una generalización del tipo anterior, donde en vez de cortar a los individuos en 1 sólo punto, se realiza en 2. Deberá tenerse en cuenta que ninguno de estos puntos de corte coincida con el extremo del cromosoma para garantizar que se originen 3 segmentos. Para ilustrar el proceso, mostramos la figura 7.3, donde tomamos los mismos padres de la figura anterior pero ahora cortando sus cromosomas en dos puntos. Note que los hijos ahora intercambian la parte central del cromosoma de los padres, permaneciendo los extremos iguales.

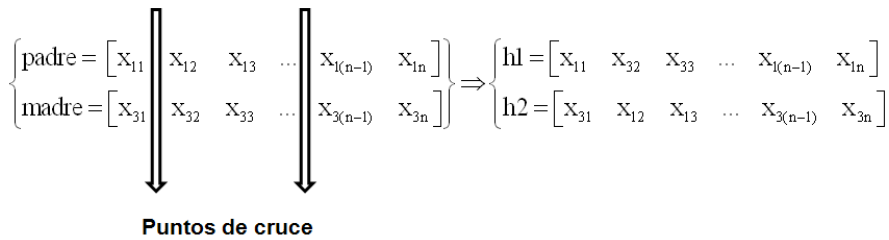


Figura 7.3: Cruce de 2 puntos

Mutación

Éste operador genético es el responsable de aumentar la diversidad en la población, permitiendo al GA desplazarse un poco más en los hiperplanos de búsqueda y evitando la convergencia prematura o, en algunos casos, el estancamiento en óptimos locales. La mutación de los individuos provoca que uno de los genes, generalmente uno por cromosoma, varíe su valor de forma aleatoria. Dichas alteraciones se pueden realizar directamente sobre los individuos antes de introducirlos en la población actual o realizarlas de manera conjunta con las operaciones de cruce. Para el último caso, se seleccionan 2 individuos y se realiza el cruce, luego se muta un gen de una de las descendencias, o ambos descendientes. De esta forma, se imita el comportamiento que se da en la naturaleza, pues cuando se generan las descendencias es posible que se produzcan fallos, generalmente sin mayor trascendencia, en el paso de la carga genética de padres a hijos.

La tasa de mutación se refiere al total de genes mutados en la población, éstas tasas suelen quedar a criterio del usuario pero normalmente son bajas, entre el 1% y 5% (Lucasius y Kateman, 1993; Michalewicz, 1996), donde autores como Haupt y Haupt (2004) atribuyen buenos resultados con 2%. Existen diversos tipos de mutaciones, Michalewicz (1996), por ejemplo, las clasifica como uniformes y no uniformes.

Mutaciones uniformes: Es la forma más simple de mutación y consiste en reemplazar totalmente el gen por un valor aleatorio dentro de su rango de dominio o alterarlo (sumar, restar o multiplicar) en una cantidad prefijada. Así por ejemplo si se ha seleccionado el individuo $(x_{r1}, x_{r2}, \dots, x_{rn})$ entonces cualquier gen x_{rk} elegido aleatoriamente ($k = 1, \dots, n$) tendrá el mismo tipo de cambio en el proceso de mutación. El resultado de este proceso sera un individuo de la forma $(x_{r1}, x_{r2}, \dots, x'_{rk}, \dots, x_{rn})$ y x'_{rk} un valor del dominio del k-ésimo parámetro.

Mutaciones no uniformes Considerado por Michalewicz (1996) como la forma más eficiente de ajustar el rendimiento del GA. Para ilustrar en qué consiste esta mutación, supongamos nuevamente el individuo $(x_{r1}, x_{r2}, \dots, x_{rn})$ y x_{rk} el gen seleccionado aleatoriamente para la mutación, tal que $x_{rk} \in [l_k, u_k]$, entonces el resultado del proceso será $(x_{r1}, x_{r2}, \dots, x'_{rk}, \dots, x_{rn})$ pero ahora

$$x'_{rk} = \begin{cases} x_{rk} + \Delta(t, u_k - x_{rk}) & \text{si sale 0} \\ x_{rk} - \Delta(t, u_k - x_{rk}) & \text{si sale 1} \end{cases}$$

donde la función $\Delta(t, y)$ retorna un valor en el rango $[0, y]$, tal que a medida que t aumente $\Delta(t, y)$ se acercará a 0. El condicional de la función corresponde al proceso aleatorio de elección que establece el usuario.

7.2. Funciones de deseabilidad

En cualquier proceso de producción, la selección del mejor conjunto de condiciones que proporcionen características y propiedades deseables en los productos se ha convertido en un problema de optimización, sobre

todo cuando se manejan simultáneamente múltiples variables respuesta como características de calidad. En situaciones más críticas, las medidas de calidad pueden tener diferente importancia, escalas de medida diferente o pueden ser medidas mutuamente opuestas, lo que equivale a dificultar aun más su manejo simultáneo. Las funciones de deseabilidad (*desirability functions*) desarrolladas por Harrington (1965) proporcionan una solución matemática a este tipo de problemas.

En efecto, Harrington (1965) sugiere funciones que combinan eficientemente las medidas de calidad mediante transformaciones que son llevadas a una escala adimensional para que cualquier propiedad o característica pueda ser interpretada en términos de calidad deseable. Dicha escala es definida como *escala de deseabilidad* y es comúnmente abreviada como d , tal que, $d = 0$ corresponde al nivel indeseable de calidad (producto inaceptable) y $d = 1$ determina el valor óptimo de calidad. Por tanto, d esta definido en el intervalo $[0, 1]$ y sus posibles valores intermedios son definidos como lo muestra la tabla 7.1.

Tabla 7.1: Escala de deseabilidad para d

d	Calidad equivalente
$d = 1,00$	Satisfacción total de calidad
$1,00 < d < 0,80$	Aceptable y excelente
$0,80 \leq d < 0,63$	Aceptable y bueno
$0,63 \leq d < 0,40$	Pobremente aceptable
$0,40 \leq d < 0,30$	En el límite de aceptación
$0,30 \leq d < 0,00$	Inaceptable
$d = 0,00$	Completamente inaceptable

Matemáticamente, d es definida en términos de sus medidas de especificación mediante la siguiente expresión:

$$d = e^{-(|z|)^n} \tag{7.3}$$

donde z es una transformación lineal de la i -ésima respuesta estimada \hat{y}_i tal que $z = -1$ cuando \hat{y}_i es igual al límite menor de especificación, $z = 1$ cuando \hat{y}_i es igual al límite mayor de especificación, n es un número positivo ($0 < n < \infty$) y $e = 2,71828$. Así pues, cualquier valor \hat{y}_i puede ser transformado a su correspondiente z_i mediante la relación:

$$z_i = \frac{2\hat{y}_i - (y^{\text{máx}} + y^{\text{mín}})}{(y^{\text{máx}} - y^{\text{mín}})}$$

donde $y^{\text{mín}}, y^{\text{máx}}$ corresponden a los valores mínimos y máximos de especificación.

La ecuación (7.3) representa una familia de curvas tal que, como se puede apreciar en la figura 7.4, se verifica:

- Es asintótica a $d = 0$ cuando $|z|$ se excede de 1
- Pasa por $d = \left(\frac{1}{e}\right) = 0,37$ cuando $|z| = 1$
- Pasa por $d = 1$ en el punto medio entre los límites de especificación inferior y superior.

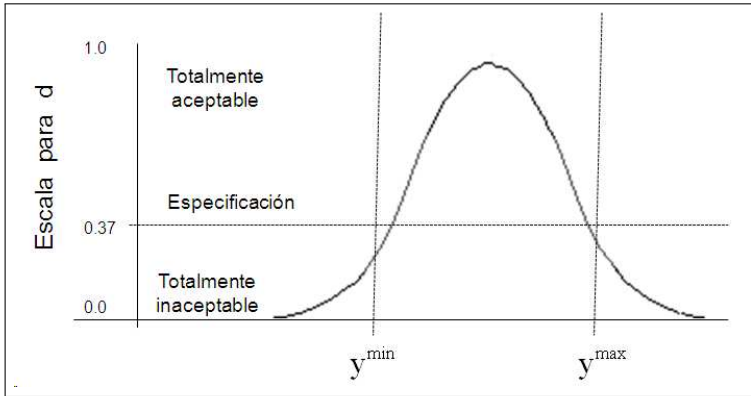


Figura 7.4: Escala de Deseabilidad

El exponente n en (7.3) determina el grado de apuntamiento de la curva (kurtosis), y puede ser calculado tomando un valor de d entre 0,60 y 0,90 y su correspondiente $|z|$ para sustituirlo en:

$$n = \frac{\ln \ln(1/d)}{\ln |z|}$$

Cuando se cuenta con un sólo valor de especificación (mínimo o máximo), la ecuación (7.3) toma la forma:

$$d = e^{-(e^{-z})} = \exp(-\exp(-z))$$

donde la pendiente de la curva es determinada por la escala de \hat{y}_i sobre z , es decir, el exponente n de la ecuación (7.3) no es requerido. Finalmente Harrington (1965) en su reporte define una medida global de calidad que combina varias medidas de calidad adimensional y la denota D . Esta medida parte de la premisa de que si un producto posee una calidad pobre tal que el producto no es adecuado para la aplicación, entonces el producto no es aceptado independientemente de las otras propiedades

de calidad en las otras aplicaciones restantes. De aquí, se define a D como la media geométrica de las j -ésimas aplicaciones de calidad consideradas en un producto, esto es:

$$D = (d_1 \cdot d_2 \cdot \dots \cdot d_j)^{\frac{1}{j}} \quad (7.4)$$

En definitiva, las funciones de deseabilidad transforman un vector de respuesta estimada \hat{Y} en una función d_j , definido como *desirability function*. Éste es un valor entre 0 y 1, el cual aumenta tanto como la deseabilidad de la respuesta correspondiente aumenta, posteriormente se unen en una función compuesta D definida como la media geométrica de los d_j . Entonces, el objetivo de optimización para el diseño de parámetros será encontrar el conjunto de variables independientes input que maximice el valor de D .

Muchas propuestas han surgido en torno a las funciones de deseabilidad exponencial entre las cuales resaltan la realizadas por Derringer y Suich (1980); Kim y Lin (2000); Hsieh y otros (2005); Wu (2005) y, más recientemente Chang (2006), quien hace una ligera modificación a la propuesta de Wu (2005). Tales propuestas tiene como factor común la clasificación de las funciones de deseabilidad según la característica de calidad del vector de respuestas estimadas como alternativas a las medidas de rendimiento propuestas por Taguchi y que definimos en el capítulo 2 del presente trabajo. Entonces, el presente trabajo adopta el enfoque de Chang (2006), quien define las funciones de deseabilidad como sigue.

Mayor es mejor (LTB):

$$d_i^{LTB} = \exp \left(- \exp \left(- \frac{\hat{y}_i - y_i^{\min}}{y_i^{\min}} \right) \right) \quad (7.5)$$

Nominal es mejor (NTB):

$$d_i^{NTB} = \exp \left(- \left| \frac{2\hat{y}_i - (y_i^{\text{máx}} - y_i^{\text{mín}})}{y_i^{\text{máx}} + y_i^{\text{mín}}} \right| \right) \quad (7.6)$$

Menor es mejor (STB)

$$d_i^{STB} = \exp \left(- \left(1 + \frac{\hat{y}_i - y_i^{\text{máx}}}{y_i^{\text{máx}}} \right) \right) \quad (7.7)$$

donde $y_i^{\text{mín}}$ y $y_i^{\text{máx}}$ corresponden a los límites de especificación inferior y superior en cada caso.

Capítulo 8

Optimización para casos univariantes

En el presente capítulo se desarrolla la fase de optimización de nuestra propuesta al problema univariante definido en (4.1), aportando así una alternativa completa y robusta para la mejora del diseño de parámetros en casos de una variable respuesta. Como se ha visto en capítulos anteriores, existen diversas propuestas para este tipo de problema, de entre las más recientes se encuentran los trabajos de Su y Chang (2000) donde emplean Simulated Annealing (SA) como método de optimización y, más recientemente, Chang (2005) quien sustituye SA por los GA en su metodología.

Nuestra propuesta, en esta fase, se enfoca en el diseño de una estrategia de optimización que combina las cualidades que ofrece los GA como método de búsqueda heurística y el algoritmo PRED-RF en la fase de modelización. En efecto, el algoritmo de modelización que hemos propuesto nos ha permitido no solo la predicción del vector de respuestas, sino que, por ser una estrategia basada en RF nos permite el uso de medidas de

importancia de variables que son incorporadas en nuestra función fitness y en las operaciones de cruce por cada generación.

El presente capítulo lo hemos dividido en 3 secciones. En la sección (8.1) definimos los componentes más importantes que controlan el proceso de nuestro GA, la sección (8.2) detalla paso a paso nuestra propuesta de optimización. Finalmente, para el análisis numérico correspondiente retomamos los casos de ilustración 4.1 y 4.2 definidos en el capítulo 4.

8.1. Componentes genéticos

Hemos definido nuestro algoritmo de optimización como un híbrido entre las cualidades que ofrece PRED-RF como estrategia de modelización y los GA como técnica de búsqueda heurística. Esto es, tomamos propiedades que nos arroja nuestra estrategia de modelización para el diseño de algunos componentes del GA, tal como explicamos a continuación.

8.1.1. Población inicial y función Fitness

El punto de partida de todo GA lo constituye la población inicial, entonces se genera aleatoriamente el conjunto de factores control \mathbf{X} dentro del intervalo de cada variable $x_p \in \mathbf{X}$ y se determinan sus valores de respuesta mediante nuestro modelo predictivo PRED-RF. Vale la pena recordar que para la creación de PRED-RF, el conjunto de datos ha sido previamente normalizado, por lo que la población inicial y sus respectivos valores respuesta serán devueltos a su escala inicial, lo que suma a nuestro algoritmo de optimización un paso adicional como se podrá ver posteriormente.

Por otra parte, PRED-RF es un algoritmo bajo el esquema de CART, por lo cual, toda nueva observación (cromosoma) será ajustada en los límites del

nodo correspondiente, lo que limita al GA en su campo de exploración. Por tal razón, definimos nuestra función fitness como una función de predicción de los nuevos cromosomas, mediante la interpolación entre nodos como, se expresa en la ecuación (8.1).

$$\hat{y}_{h_i} = \left| \frac{\hat{y}_{m_i}(g_{p_i} - g_{h_i}) - \hat{y}_{p_i}(g_{m_i} - g_{h_i})}{g_{p_i} - g_{m_i}} \right| \quad (8.1)$$

donde \hat{y}_{h_i} representa la estimación de la respuesta del i -ésimo cromosoma hijo, \hat{y}_{m_i} y \hat{y}_{p_i} la predicción de la i -ésima madre y el i -ésimo padre provenientes de PRED-RF. Finalmente g_{p_i} , g_{m_i} y g_{h_i} corresponden a los valores del gen del i -ésimo padre, i -ésima madre e i -ésimo hijo, que serán determinados por las medidas de importancia de las variables proveniente de PRED-RF, como explicamos a continuación.

Nuestro algoritmo PRED-RF provee de medidas de importancia de las variables, lo que usaremos para dar pesos a los genes y de esta forma ayudar al GA en su búsqueda de forma más dirigida. Así, aquel gen que obtenga el mayor peso será el usado en la ecuación (8.1). Dichos pesos serán determinados mediante la ecuación (8.2) y serán usados posteriormente en las operaciones de cruce.

$$PC = \frac{I_{x_k}}{\sum_{k=1}^p I_{x_k}} \quad (8.2)$$

donde I_{x_k} representa la importancia del k -ésimo gen en PRED-RF, ($k = 1, \dots, p$).

8.1.2. Cruce ponderado

Para la formación de las siguientes generaciones se adopta el cruce simple de 1 punto pero con una variante: la asignación de pesos a los genes del

vector de factores control \mathbf{X} , mediante la medida de importancia de las variables calculadas en (8.2). De esta forma, aumentamos la probabilidad de cruce de los cromosomas en torno a los genes de mayor importancia en el diseño.

8.2. Estrategia de optimización

La presente sección tiene como objetivo detallar los pasos de nuestra propuesta de optimización, donde se incluyen los componentes definidos anteriormente. Dicha propuesta, basada en GA, definirá su objetivo dependiendo de la característica de calidad considerada en los respectivos casos de estudio con el fin de determinar la combinación óptima de los niveles del factor control \mathbf{X} como espacio de solución, en respuesta al problema planteado mediante la ecuación (4.1).

El proceso se inicia formando aleatoriamente n cromosomas. Los cromosomas están formados por $p = k + j + l$ genes, que corresponden a los k elementos del vector de factores de control, j niveles del factor señal y l niveles del factor ruido. Denotaremos a las poblaciones como a_z , tal que la población inicial $z = b$ es denotada como $a_z = a_b$, donde a_{bi} representa el i -ésimo cromosoma de la población b .

La nueva generación estará formada por la unión de los cromosomas padres a_{bi} y los hijos a'_{bi} con sus correspondientes valores respuesta \hat{y}_i . Esto es $a_{Bi} = a_{bi} \cup a'_{bi}$.

Como condición de parada se comparan las respuestas estimadas de la generación actual, a_{Bi} , con la inicial, a_b , en consideración a la característica de calidad bajo estudio y la convergencia de los cromosomas en un mismo

rango de respuestas.

Inicialización. Hacer $z=b$

- Emplear PRED-RF para determinan las respuestas de la población inicial a_b .
- Aplicar la ecuación (8.2) para la asignación del peso de importancia a cada gen en el vector \mathbf{X} .
- Llevar el conjunto de cromosomas con sus correspondientes predicciones a escala inicial. Así queda conformada la población inicial a_b .
- Definir la tasa de mutación t y el criterio de selección de los individuos más aptos.

Paso 1: Evaluación. Determinar el valor de respuesta y_i más cercana a la característica de calidad e ir al paso 2.

Paso 2: Selección. Seleccionar los individuos más aptos de acuerdo al criterio definido en el proceso de inicialización. A esta selección la definimos como a_{bi}^i .

- **Cruce:** Determinar aleatoriamente el gen del vector \mathbf{X} como punto de cruce y realizar cruce en a_{bi}^i .
- **Mutación:** Llevar a cabo la mutación de los genes mediante la tasa t definida en el proceso de inicialización.
- **Predicción para la nueva generación:** $\forall a_{bi}^i$ determinar el valor de respuesta \hat{y}_i mediante la ecuación (8.1).

Paso 3: Reemplazo. Hacer $a_{Bi} = a_{bi} \cup a_{bi}^i$.

Paso 4: Convergencia :

4.1 Comparar el valor de respuesta y_i del paso 1 con las respuestas \hat{y}_i , ¿es \hat{y}_i un óptimo, $\forall \hat{y}_i \in a_{B_i}$?. En caso afirmativo ir a 4.2, en caso contrario ir a paso 2.

4.2 ¿ $a_{z=B}$ converge? en caso afirmativo ir al paso 5, caso contrario hacer $z = b + 1$ e ir a paso 2.

Paso 5: Obtener la combinación óptima de los niveles de los parámetros en \mathbf{X} de la respuesta \hat{y}_i .

8.3. Casos de ilustración

La presente sección tiene como objetivo ilustrar el algoritmo de optimización diseñado en las secciones anteriores como punto final de nuestra propuesta metodológica al problema del diseño de parámetros univariante. A tal fin, retomamos los casos 4.1 y 4.2 tratados en el capítulo 4. Se comparan los resultados obtenidos con los trabajos de Su y Chang (2000); Chiu y otros (1997) en el caso 1 y con Chang (2005) para el caso 2.

8.3.1. Optimización del caso 4.1

Componentes genéticos

Asumimos a PRED-RF como algoritmo de modelización bajo la función de normalización f_2 de la tabla 4.1, en vista de los resultados obtenidos en la sección (4.4.2). Se estiman las respuestas de 400 cromosomas generados aleatoriamente con la estructura del vector $\mathbf{X} = \{A, B, C, D, E, F, G, H\}$ en escala normalizada. Por otra parte, se calculan los pesos de cada gen mediante la función (8.2) donde I_{x_k} representa, en porcentaje, el valor de importancia del k -ésimo factor (gen) en PRED-RF como lo muestra la tabla 8.1. En dicha tabla, mostramos además, el peso asignado a cada gen para las operaciones de cruce.

Tabla 8.1: Medidas de importancia y pesos de los genes según PRED-RF.
Caso 4.1

Gen	A	B	C	D	E	F	G	H
I_{x_k}	10.134	28.601	18.318	16.1313	17.810	64.761	40.125	20.995
Pesos	0.047	0.132	0.084	0.074	0.082	0.299	0.185	0.097

La tabla anterior también nos permite determinar nuestra función fitness. Como se puede observar, F es el gen con mayor peso, por lo que la ecuación (8.1) queda denotada como sigue

$$\hat{y}_{h_i} = \left| \frac{\hat{y}_{m_i}(f_{p_i} - f_{h_i}) - \hat{y}_{p_i}(f_{m_i} - f_{h_i})}{f_{p_i} - f_{m_i}} \right| \quad (8.3)$$

donde f_{p_i} , f_{m_i} y f_{h_i} corresponden a los valores del gen F del i-ésimo padre, i-ésima madre e i-ésimo hijo correspondientemente. Entonces, nuestro algoritmo de optimización queda estructurado como lo muestra la sección siguiente. Cabe resaltar que dicho algoritmo, así como todas las operaciones del presente trabajo han sido procesadas y codificadas en lenguaje R.

Implementación de la estrategia de optimización al caso 4.1

Presentamos a continuación la aplicación de la estrategia de optimización, establecida en la sección 8.2, al caso de ilustración modelizado en el apartado 4.4.2.

Inicialización. Determinar aleatoriamente 400 cromosomas, formados por 8 genes, correspondientes a $\mathbf{X} = \{A, B, C, D, E, F, G, H\}$ en escala normalizada.

- Hacer $z=b$
- Hacer $PRED - RF(a_b)$ para determinan las \hat{y}_{a_b} .

-
- La asignación de los pesos de cada gen en el vector \mathbf{X} queda determinada por la tabla 8.1.
 - Llevar el conjunto de cromosomas con sus correspondientes predicciones a escala inicial. Así queda conformada la población inicial a_b .
 - Se define como tasa de mutación $t = 2\%$. y como criterio de selección los individuos cuyos valores y_i sean menores que la mediana en la población b ; es decir $a'_{bi} = \{a_{bi}; y_{bi} \leq \tilde{y}_b\}$.

Paso 1: Evaluación. Determinar $\min(y_i)$ con su correspondiente combinación en el vector X .

Paso 2: Selección. Calcular $a'_{bi} = \{a_{bi}; y_{bi} \leq \tilde{y}_b\}$

- **Cruce:** Realizar cruce en a'_{bi} .
- **Mutación:** Llevar a cabo con $t = 2\%$.
- **Predicción para la nueva generación:** $\forall a'_{bi}$ determinar \hat{y}_i mediante la ecuación (8.3).

Paso 3: Reemplazo. Hacer $a_{Bi} = a_{bi} \cup a'_{Bi}$.

Paso 4: Convergencia :

- 4.1 Calcular $\min(\hat{y}_{a_{Bi}})$. Si $\min(\hat{y}_{a_{Bi}}) \leq \min(\hat{y}_{a_{bi}})$ ir a 4.2. En caso contrario, hacer $z = b + 1$ e ir a paso 2.
- 4.2 ¿ $a_{z=B}$ converge?, en caso afirmativo ir al paso 5 y culminar. Caso contrario, hacer $z = b + 1$ e ir a paso 2.

Paso 5: Obtener la combinación óptima de los niveles del factor control \mathbf{X} para $\min(\hat{y}_{a_{Bi}})$.

Los resultados de nuestro algoritmo de optimización para 17 generaciones se presentan en la figura 8.1, donde se puede ver una clara convergencia a partir de la generación 11. Seguidamente, en la tabla 8.2 comparamos nuestro resultado con los obtenidos por Su y Chang (2000) y Chiu y otros (1997), así como con el resultado obtenido por la metodología Taguchi, (mostrado en Su y Chang, 2000).

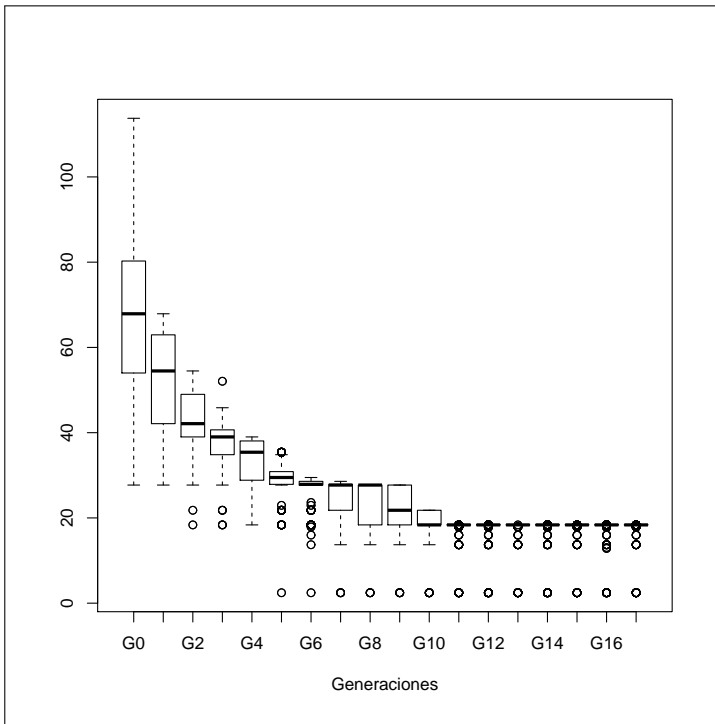


Figura 8.1: Comportamiento del GA para la longitud del canal de gas. Caso 4.1

Tabla 8.2: Comparación de los resultados en diferentes propuestas de optimización. Caso 4.1.

Método	Combinación de los parámetros								\hat{y}
	A	B	C	D	E	F	G	H	
Taguchi	50	240	50	2	130	64	1	3	19.8
Chiu y otros (1997)	50	240	50	2	130	63.5	1	6	13.5
Su y Chang (2000)	48.2	235	46	0.85	85.1	64	1	6	7.4
Nuestra propuesta	50.16	235.02	50.96	0.24	91.44	64.16	0.25	0	2.45

Se puede observar cómo los resultados por el método Taguchi y la propuesta por Chiu y otros (1997), basada en ANN, son ligeramente diferentes en la combinación de los parámetros pero menos ligeras en la respuesta estimada. Diferencias mucho más notorias se muestran en la propuesta de Su y Chang (2000), basada en SA, con una estimación casi a la mitad del obtenido por Chiu y otros (1997). Finalmente, se presentan nuestros resultados, donde no sólo reducimos la respuesta estimada en aproximadamente 5 puntos, sino que una de las diferencias más notorias en la combinación de los parámetros es presentada por el valor de H, quien corresponde a uno de los factores con mayor peso de importancia en la tabla 8.1, corroborando nuevamente la eficiencia de nuestro algoritmo de modelización.

8.3.2. Optimización del caso 4.2

Componentes genéticos

El presente caso de ilustración corresponde a la continuación del caso 4.2 estudiado en la sección (4.4.3), donde la característica de calidad es del tipo estático NTB y se busca ajustar la media a un punto o valor nominal con la existencia de tolerancias bilaterales. El objetivo consiste en determinar los niveles de los factores control para hacer la respuesta cercana al valor 0.2285mm. Por lo tanto, el criterio de selección de los individuos más

aptos lo hemos definido como los cromosomas cuyas respuestas estimadas pertenezcan al intervalo $[0.18, 0.26]$.

Los cromosomas están compuestos por 7 genes: correspondientes a los 6 factores de control que componen \mathbf{X} y el factor de ruido. Sus respectivas respuestas son estimadas mediante el algoritmo PRED-RF ejecutado en el apartado (4.4.3). Considerando que el objetivo es determinar la combinación óptima de los niveles de los parámetros, se determinan la importancia de los genes pertenecientes a \mathbf{X} de acuerdo a la función (8.2), donde I_{x_k} representa el porcentaje de importancia del k-ésimo gen en PRED-RF. Todo lo anterior se muestra en la tabla 8.3. En dicha tabla mostramos, además, el peso asignado a cada gen para las operaciones de cruce.

Tabla 8.3: Medidas de importancia y pesos de los genes según PRED-RF. Caso 4.2

Gen	A	B	C	D	E	F
I_{x_k}	62.937	45.115	33.680	36.658	53.054	40.532
Pesos	0.233	0.160	0.124	0.136	0.197	0.150

La tabla anterior muestra al gen A como el de mayor importancia, entonces, de acuerdo a la ecuación (8.1), definimos la función fitness en este caso como sigue

$$\hat{y}_{h_i} = \left| \frac{\hat{y}_{m_i}(A_{p_i} - A_{h_i}) - \hat{y}_{p_i}(A_{m_i} - A_{h_i})}{A_{p_i} - A_{m_i}} \right| \quad (8.4)$$

donde A_{p_i} , A_{m_i} y A_{h_i} corresponden a los valores del gen A del i-ésimo padre, i-ésima madre e i-ésimo hijo respectivamente. Nuestro algoritmo de optimización queda estructurado como lo muestra la sección siguiente.

Implementación de la estrategia de optimización al caso 4.2

Presentamos a continuación la aplicación de la estrategia de optimización, establecida en la sección 8.2, al caso de ilustración modelizado en el apartado 4.4.3.

Inicialización. Determinar aleatoriamente 100 cromosomas, formados por 7 genes, correspondientes a $\mathbf{X} = \{A, B, C, D, E, F\}$, en escala normalizada, y el factor ruido en los niveles $\{N_1, N_2\}$.

- Hacer $z=b$
- Hacer $PRED - RF(a_b)$ para determinar las \hat{y}_{a_b} .
- La asignación de los pesos de cada gen en el vector \mathbf{X} queda determinada por la tabla 8.3
- Llevar el conjunto de cromosomas con sus correspondientes predicciones a escala inicial. Así queda conformada la población inicial a_b .
- Se define como tasa de mutación $t = 2\%$, y como criterio de selección los individuos cuyos valores $y_i \in [0.18, 0.26]$; es decir $a_{bi}^j = \{a_{bi}; y_{bi} \in [0.18, 0.26]\}$.

Paso 1: Evaluación. Determinar el (y_i) más cercano a 0.2285 con su correspondiente combinación en el vector X .

Paso 2: Selección. Calcular $a_{bi}^j = \{a_{bi}; y_{bi} \in [0.18, 0.26]\}$

- **Cruce:** Realizar cruce $\forall a_{bi}^j$.
- **Mutación:** Llevar a cabo con $t = 2\%$.
- **Predicción para la nueva generación:** $\forall a_{bi}^j$ determinar \hat{y}_i mediante la ecuación (8.4).

Paso 3: Reemplazo. Hacer $a_{Bi} = a_{bi} \cup a'_{Bi}$.

Paso 4: Convergencia :

4.1 Calcular $\hat{y}_{a_{Bi}}$ más cercano a 0.2285.

Si $\hat{y}_{a_{Bi}} \neq \hat{y}_{a_{bi}}$ y $(\hat{y}_{a_{Bi}} - 0.2285) \rightarrow 0$ ir a 4.2, en caso contrario hacer $z = b + 1$ y volver al paso 2.

4.2 ¿ $a_{z=B}$ converge?, en caso afirmativo ir al paso 5 y culminar.

Caso contrario, hacer $z = b + 1$ e ir al paso 2.

Paso 5: Obtener la combinación óptima de los niveles del factor control \mathbf{X} para $\hat{y}_{a_{Bi}}$.

El comportamiento de nuestro algoritmo de optimización para 28 generaciones se presenta en la figura 8.2. El objetivo del presente caso de ilustración es acercar el valor de la respuesta a 0.2258. Se puede notar a partir de la generación 21 cómo las cajas minimizan su tamaño en valores cercanos a 0.2, lo que indica la convergencia cercana al valor objetivo.

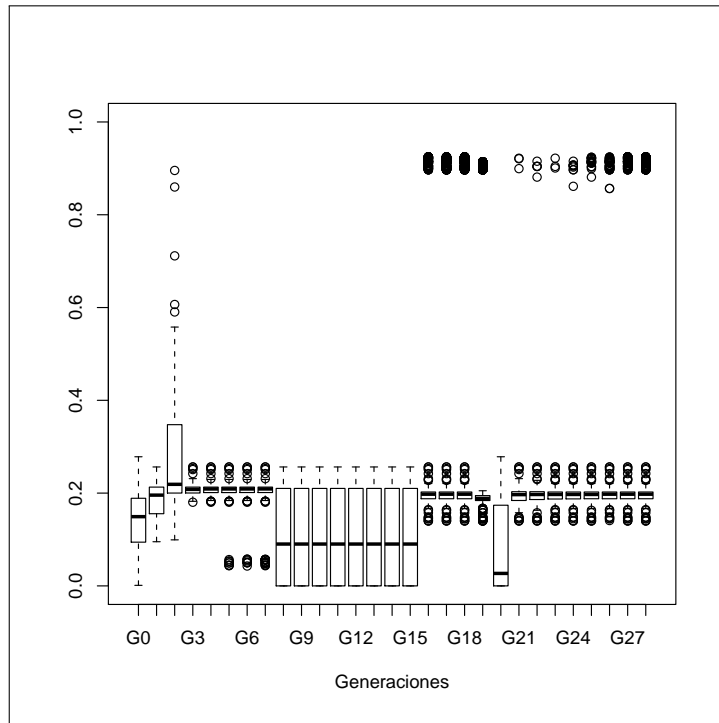


Figura 8.2: Comportamiento del GA para el grosor del patrón epoxy.
Caso 4.2

En la tabla 8.4 comparamos nuestro resultado con el obtenido por Chang (2005), así como con el resultado obtenido bajo la metodología Taguchi, (mostrado en Chang, 2005). Se puede ver cómo los mejores resultados son alcanzados por nuestra metodología propuesta y la de Chang (2005), mejorando el valor de la respuesta en casi un 50% con respecto a la metodología Taguchi. En cuanto a las diferencias entre nuestra metodología y la de Chang (2005), se notan leves discrepancias en los resultados de \hat{y} , pero diferencias más marcadas se reflejan en los niveles de los factores D y E, factores que poseen los segundos valores más altos en importancia según nuestro esquema de modelaje PRED-RF y que reflejamos en la tabla 8.3.

Tabla 8.4: Comparación de los resultados en diferentes propuestas de optimización. Caso 4.2

Método	Combinación de los parámetros						\hat{y}
	A	B	C	D	E	F	
Taguchi	0.032	3	0.02	60	0.02	100	0.4207
Chang (2005)	0.044	2.9	0.057	12	0.022	118	0.2286
Nuestra propuesta	0.04	2.9	0.021	39.04	0.012	123.8	0.2275

Para finalizar el presente capítulo, resaltamos que los 2 casos de ilustración estudiados no sólo reflejan la eficiencia de nuestra metodología propuesta en cuanto a los objetivos de calidad establecidos en cada caso, sino que hemos podido constatar su eficiencia en problemas con estructura más compleja y con objetivos de calidad de tipo estático, lo que amplía su campo de aplicación.

Capítulo 9

Optimización para casos multivariantes

El presente capítulo corresponde a la fase de optimización de nuestra propuesta al problema multi-respuesta definido en (5.1) como una generalización del caso univariante, aportando así una alternativa completa y robusta para la mejora del diseño de parámetros en casos de múltiples variables respuestas. Como se ha visto en capítulos anteriores, existen diversas propuestas para este tipo de problema donde el factor común es el empleo de las funciones exponenciales de deseabilidad definidas por Harrington (1965). Entre los trabajos más recientes se encuentra el presentado por Chang y Chen (2011), donde no sólo incorpora dichas funciones sino que propone la fase de optimización mediante el uso de algoritmos genéticos en virtud a los beneficios que como método de optimización heurística posee.

Nuestra propuesta de optimización se basa en un híbrido entre MRF, definido en el apartado anterior, las funciones de deseabilidad y los GA. En efecto, hemos integrando las funciones de deseabilidad exponencial a

los GA y, a diferencia de Chang y Chen (2011), incorporamos variaciones importantes en los procesos de cruce y evaluación de los cromosomas, provenientes de nuestro esquema de modelización MRF. Análogamente al caso univariante, nuestra propuesta de modelización nos ha permitido no sólo la predicción del vector de respuestas, sino que, por ser una estrategia basada en RF nos permite el uso de medidas de importancia de las variables que son incorporadas en las operaciones de cruce por cada generación. Además, hemos diseñado en el proceso de reproducción de los cromosomas, una medida de ajuste de la nueva generación que permite, mediante interpolaciones, explorar valores de predicción entre nodos con el fin de propiciar más diversidad en la población de búsqueda.

Presentamos nuestro algoritmo de optimización en 3 secciones. En la sección (9.1) definimos los componentes que controlan el proceso del GA en todas sus etapas, en la sección (9.2) se integran todos los componentes de las secciones anteriores y se detalla paso a paso nuestra propuesta de optimización. Finalmente, en la sección (9.3) se realizan los análisis numéricos de las fortalezas y debilidades de nuestra propuesta mediante dos casos de ilustración, contrastándose con las propuestas aportadas por Chang (2006, 2008) y posteriormente con Chang y Chen (2011).

9.1. Componentes genéticos

Hemos definido nuestro algoritmo de optimización como un híbrido entre las cualidades que ofrece nuestro MRF como estrategia de modelización, las funciones de deseabilidad como estrategia que permite el estudio de diferentes medidas de calidad simultáneamente y GA como técnica de búsqueda heurística. Esto es, tomamos propiedades que nos arroja nuestra estrategia de modelización y las funciones de deseabilidad exponencial

para el diseño de algunos componentes del GA, tal como explicamos a continuación.

9.1.1. Población inicial y función Fitness

Nuevamente, el punto de partida del GA lo constituye la población inicial. Entonces se genera una población aleatoria, cuyo valor de predicción proviene de MRF_{Pp} . Análogamente al caso univariante, acotamos que en la creación de nuestro modelo predictivo MRF_{Pp} , el conjunto de datos ha sido previamente normalizado, por lo que, tanto la población inicial como su predicción, serán devueltas a su escala inicial. Además, para la integración del vector de respuestas $\mathbf{Y} = \{y_1, y_2, \dots, y_r\}$ en una medida global hemos empleado las funciones definidas en la tabla 5.1 por lo tanto, y usando los correspondientes pesos de cada variable respuesta, se deben realizar las r estimaciones pertinentes. Todo lo anterior, suma a este algoritmo de optimización dos pasos en su inicio, los cuales serán detallados en la sección 9.2.

Función fitness

Un sistema dinámico puede ser considerado como un sistema multi-target de tipo estático (Joseph y Wu, 2002). Esto nos permite establecer las funciones de deseabilidad definidas en (7.5), (7.6) y (7.7) como nuestras medidas de rendimiento (PMs, por sus siglas en inglés) acorde a los respectivos tipos de respuesta.

Tabla 9.1: Medidas de rendimiento según el tipo de respuesta

$\text{DLTB: } d_i = \exp \left(- \exp \left(- \frac{1}{sn} \sum_{j=1}^s \sum_{l=1}^n \frac{\hat{y}_{ijkl} - y_{ij}^{\min}}{y_{ij}^{\min}} \right) \right)$
$\text{DNTB: } d_i = \exp \left(- \frac{1}{sn} \sum_{j=1}^s \sum_{l=1}^n \left \frac{2\hat{y}_{ijkl} - (y_{ij}^{\max} + y_{ij}^{\min})}{y_{ij}^{\max} + y_{ij}^{\min}} \right \right)$
$\text{DSTB: } d_i = \exp \left(- \left(1 + \frac{1}{sn} \sum_{j=1}^s \sum_{l=1}^n \frac{\hat{y}_{ijkl} - y_{ij}^{\max}}{y_{ij}^{\max}} \right) \right)$

donde y_{ij}^{\min} , y_{ij}^{\max} representan los límites inferior y superior de especificación de la i -ésima respuesta en el j -ésimo nivel del factor señal.

Harrington (1965) define un índice de rendimiento total para evaluar el rendimiento de las respuestas de calidad de forma simultánea. Tal índice se define como la media geométrica de las respuestas de calidad en su correspondiente función de deseabilidad como se presenta en (9.1) y que denotamos como OPI (Overall Performance Index, Chang (2008)), para ser incorporado en nuestro GA como función de evaluación fitness.

$$OPI = \left(\prod_{j=1}^r d_j \right)^{1/r} \quad (9.1)$$

donde d_j representa el valor de deseabilidad en el j -ésimo vector de respuesta, con $(j = 1, \dots, r)$

9.1.2. Cruce ponderado

Para la formación de las siguientes generaciones, se adopta el cruce de un punto pero nuevamente con la variante de la asignación de pesos a los genes del vector de factores control \mathbf{X} mediante la medida de importancia de las variables, en este caso arrojada por MRF_{Pp} . De esta forma, aumentamos la probabilidad de cruce de los cromosomas en torno a los genes de mayor importancia en el diseño. Se define el peso de cada gen mediante la ecuación siguiente, donde en este caso I_{x_k} representa la importancia del k -ésimo gen en MRF_{Pp} , ($k = 1, \dots, p$).

$$PC = \frac{I_{x_k}}{\sum_{k=1}^p I_{x_k}} \quad (9.2)$$

9.1.3. Funciones de mutación

En cuanto a la tasa y tipo de mutación empleada, proponemos un abanico de posibilidades considerando la importancia teórica que la mutación posee en los GA en general. En primera instancia, proponemos en el estudio el 2% y el 5% como tasa de mutación, lo que influye directamente en el número de genes mutados en cada generación. Seguidamente estudiamos la posibilidad de que el mismo gen pueda o no ser mutado en diferentes generaciones, lo que hemos calificado como mutación con o sin reemplazamiento. Finalmente, proponemos 4 funciones $m(g)$ de mutación que determinan el campo de búsqueda del algoritmo. Las funciones se presentan en la tabla 9.2, donde g representa el gen a mutar en el cromosoma correspondiente y g_{min} , g_{max} los valores mínimo y máximo del gen en la población.

En dicha tabla se puede observar cómo las dos primeras funciones $m_1(g)$, $m_2(g)$ son mutaciones no uniformes pero diseñadas tal que alteran

el gen dentro del rango de \mathbf{X} , mientras que $m_3(g)$ y $m_4(g)$ son mutaciones uniformes pero más abiertas que podrían permitir, en algunos casos, que el algoritmo pueda hacer búsquedas fuera del rango de \mathbf{X} .

Tabla 9.2: Funciones de mutación para el caso multivariante

$m_1(g) = \begin{cases} g + g(0,10) & g < g_{max} \\ g & \text{en otros casos} \end{cases}$
$m_2(g) = \begin{cases} g - g(0,10) & g > g_{min} \\ g & \text{en otros casos} \end{cases}$
$m_3(g) = g + 1$
$m_4(g) = g - 1$

En resumen, proponemos 2 tasas de mutación, consideradas con y sin reemplazo, en 4 funciones diferentes, lo que puede ser expresado, a fines de análisis posteriores, como un modelo factorial de efectos fijos tal como lo expresa la ecuación (9.3). Donde la respuesta corresponde al valor OPI del i -ésimo cromosoma considerando la t -ésima tasa de mutación ($t = 2\%, 5\%$), m -ésima función de mutación ($m = m_1, m_2, m_3, m_4$), definidas en la tabla 9.2, y el q -ésimo factor de reemplazo, ($q = S, N$).

$$OPI_{imtr} = \eta_t + \tau_m + \gamma_q + (\eta\tau)_{tm} + (\eta\gamma)_{tq} + (\tau\gamma)_{mq} \quad (9.3)$$

Con ello se logrará obtener numéricamente el mejor esquema de mutación

en nuestro algoritmo de optimización, mediante un ANOVA.

9.1.4. Predicción de nuevos cromosomas

Luego de las operaciones de cruce y mutación es necesaria la predicción del valor respuesta de los nuevos cromosomas. Ahora bien, nuestro modelo predictivo definido como MRF_{Pp} tiene sus bases en CART, por lo que cualquier nueva observación (cromosoma) será ajustado en los límites del nodo correspondiente. Esto, nuevamente, limita al GA en su tarea exploradora. Entonces, considerando que la operación de cruce corresponde a un intercambio genético entre cromosomas y que en nuestro modelo MRF_{Pp} los genes están representados por nodos, asignamos como valor de respuesta a los nuevos cromosomas la interpolación entre dichos nodos en el gen de mayor peso del cromosoma.

En términos matemáticos, todo lo anterior se resume en la ecuación (8.1) que presentamos nuevamente a continuación, pero en este caso, \hat{y}_{m_i} e \hat{y}_{p_i} representan la predicción de la i -ésima madre y el i -ésimo padre provenientes de MRF_{Pp} . Finalmente, g_{p_i} , g_{m_i} y g_{h_i} corresponden a los valores del gen del i -ésimo padre, i -ésima madre e i -ésimo hijo con mayor peso en el modelo MRF_{Pp} .

$$\hat{y}_{h_i} = \left| \frac{\hat{y}_{m_i} (g_{p_i} - g_{h_i}) - \hat{y}_{p_i} (g_{m_i} - g_{h_i})}{g_{p_i} - g_{m_i}} \right|$$

9.2. Estrategia de optimización

La presente sección tiene como objetivo detallar los pasos de nuestra propuesta de optimización, en donde se incluyen los componentes definidos anteriormente. Dicha propuesta, orientada en un GA, tiene como objetivo maximizar el valor OPI con el fin de determinar la combinación óptima de

los niveles del factor control \mathbf{X} como espacio de solución, en respuesta al problema planteado mediante la ecuación (5.1).

El proceso se inicia formando aleatoriamente n cromosomas. Los cromosomas están formados por $(2 + p)$ genes: el factor señal $M_j, (j = 1, \dots, s)$, el factor de ruido, $Z_l, (l = 1, \dots, n)$, y el vector de k factores de control $X_k, (k = 1, \dots, p)$ en escala normalizada. Denotaremos a las poblaciones como a_z , tal que la población inicial $z = b$ es denotada como $a_z = a_b$. Así, a_{bi} representa el i -ésimo cromosoma de la población b .

Como criterio de selección en cada a_{bi} se emplea la mediana OPI, que denotaremos como \widetilde{OPI} , y se seleccionan todos los cromosomas cuyos valores OPI sean mayores a \widetilde{OPI} . A tal selección la denotamos como a'_{bi} tal que $a'_{bi} = \{a_{bi}; OPI(a_{bi}) > \widetilde{OPI}\}$.

La nueva generación estará formada por la unión de los cromosomas padres a_{bi} y los hijos a'_{bi} con su correspondiente vector $\hat{\mathbf{Y}}$. Esto es $a_{Bi} = a_{bi} \cup a'_{bi}$. Como condición de parada se comparan las respuestas estimadas de la generación actual a_{Bi} con la inicial a_b , en consideración al OPI y la convergencia de los cromosomas en un mismo rango de respuestas.

Inicialización. Hacer $z=b$

- Emplear MRF_{Pp} para determinan los valores predictivos del vector $\hat{\mathbf{Y}} = (\hat{y}_1, \dots, \hat{y}_r)$.
- Aplicar la ecuación (8.2) para la asignación del peso de importancia a cada gen en el vector \mathbf{X} .
- Llevar el conjunto de cromosomas con sus correspondientes predicciones a escala inicial. Así queda conformada la población inicial a_b .

Paso 1: Evaluación. Calcular los PMs correspondientes a cada tipo de respuesta del vector $\hat{\mathbf{Y}}$ según la tabla 9.1 y la medida OPI con la ecuación (9.1).

Paso 2: Selección. Calcular \widetilde{OPI} y determinar $a'_{bi} = \{a_{bi}; OPI(a_{bi}) > \widetilde{OPI}\}$.

- **Cruce:** Determinar aleatoriamente el factor del vector \mathbf{X} como punto de cruce y realizar cruce en a'_{bi} .
- **Mutación:** Emplear la función $m(g)$ (tabla 9.2) para la mutación de los genes, con tasa de mutación t y el factor q de reemplazo en todo el conjunto a'_{bi} .
- **Predicción para la nueva generación:** $\forall a'_{bi}$ determinar el vector $\hat{\mathbf{Y}}$ de respuestas mediante la ecuación (8.1).

Paso 3: Reemplazo. Hacer $a_{Bi} = a_{bi} \cup a'_{bi}$.

Paso 4: Convergencia. Calcular $OPI_{a_{Bi}}$.

4.1 Si $\max[OPI_{a_{Bi}}] \leq OPI_{a_{bi}}$, hacer $z = b + 1$ y volver al paso 2. En caso contrario ir a 4.2.

4.2 ¿ $a_{z=B}$ converge?, en caso afirmativo, ir al paso 5 y culminar. Caso contrario hacer $z = b + 1$ e ir al paso 2.

Paso 5: Obtener la combinación óptima de los niveles del factor control \mathbf{X} y su correspondiente multi-respuesta.

9.3. Casos de ilustración

Para finalizar el presente capítulo y en concordancia con nuestra propuesta de modelización, se retorna a los 2 casos de ilustración modelados en el capítulo 5 (secciones 5.4.2 y 5.4.3 respectivamente). Nuevamente, y con la finalidad de realizar análisis numéricos exhaustivos de las virtudes y

deficiencias de nuestra propuesta, empleamos el primer caso de estudio como un análisis piloto en la determinación de los componentes del GA. Es decir, mediante el primer caso de estudio (punto 9.3.1) determinamos un esquema de mutación que posteriormente será heredado al segundo caso de estudio en el punto 9.3.2. Cabe resaltar que en cada ilustración se realizan comparaciones numéricas con los trabajos de Chang (2006, 2008) y Chang y Chen (2011) respectivamente.

9.3.1. Optimización del caso 5.1

Establecimiento de los componentes del AG

Población inicial y función Fitness

Recordemos que este caso de estudio corresponde a un diseño factorial de tipo $2^1 \cdot 3^7$ dispuesto en un arreglo ortogonal L_{18} . Pero si consideramos dichos factores fuera de la matriz L_{18} , tendríamos un número exagerado de posibles combinaciones de los factores lo cual, en casos industriales, resulta sumamente costoso. Por otra parte, recordemos que una de las ventajas de nuestro algoritmo de modelización es la robustez de predicción de nuevos individuos y que los GA soportan en su proceso grandes poblaciones. Por todo lo anterior, decidimos generar aleatoriamente 400 cromosomas, cuya estructura está conformada por un total de 8 genes; 6 genes corresponden al vector de factores control $\mathbf{X} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, 1 gen corresponde al factor señal $M = \{10, 20, 30\}$ y el último gen corresponde al factor de ruido $Z = \{N_1, N_2\}$. Dichos cromosomas estarán en escala normalizada a fin de calcular sus correspondientes respuestas con nuestro modelo predictivo MRF_{P_2} . De esta forma queda conformada la población inicial para la inicialización del algoritmo.

El objetivo del presente caso de estudio es la optimización simultánea del

vector de respuestas $\mathbf{Y} = \{y_1, y_2, y_3\}$ bajo las características de calidad DLTB, DNTB y DSTB respectivamente, la cuales fueron definidas en el capítulo 2. Las funciones de deseabilidad definidas en la tabla 9.1 y la correspondiente medida de rendimiento OPI (ecuación (9.1)) queda determinada por la expresión (9.4), y representará la función *fitness* del algoritmo para este caso.

$$\begin{aligned}
 OPI = & \left\{ \exp \left[- \exp \left(- \frac{1}{6} \sum_{j=1}^3 \sum_{l=1}^2 \frac{\hat{y}_{1jkl} - \hat{y}_{1j}^{\min}}{\hat{y}_{1j}^{\min}} \right) \right] \right. \\
 & \cdot \exp \left(- \frac{1}{6} \sum_{j=1}^3 \sum_{l=1}^2 \left| \frac{2\hat{y}_{2jkl} - (y_{2j}^{\max} + y_{2j}^{\min})}{y_{2j}^{\max} - y_{2j}^{\min}} \right| \right) \\
 & \cdot \left. \exp \left[- \left(1 + \frac{1}{6} \sum_{j=1}^3 \sum_{l=1}^2 \frac{\hat{y}_{3jkl} + y_{3j}^{\max}}{y_{3j}^{\max}} \right) \right] \right\}^{\frac{1}{3}} \quad (9.4)
 \end{aligned}$$

Cruce ponderado

Análogamente al caso univariante, nuestro algoritmo de modelización nos permite conocer la medida de importancia de las variables en dicho modelaje, lo cual empleamos para asignar pesos a los genes que conforman los cromosomas. Cabe destacar que los cromosomas están conformados por el vector \mathbf{X} , S y Z , pero que el proceso de optimización se concentra en encontrar la combinación de los factores de \mathbf{X} que maximicen el valor OPI; por tanto, las operaciones de cruce y mutación sólo serán realizadas a los genes de \mathbf{X} como espacio de búsqueda heurística.

Presentamos la tabla 9.3, donde se reflejan los valores de importancia ($I_{xk}, k = 1, \dots, 6$) y los pesos asignados a los genes para los cruces calculados mediante la ecuación (9.2). Se puede notar que es x_2 el gen con mayor

valor de importancia, lo cual determinará posteriormente la definición de la función de predicción para los nuevos cromosomas en el presente caso ilustrativo.

Tabla 9.3: Medidas de importancia y pesos de los genes según MRF_{P2} .
Caso 5.1

Gen	x_1	x_2	x_3	x_4	x_5	x_6
I_{xk}	0.403	0.602	0.398	0.464	0.436	0.459
Pesos	0.146	0.218	0.144	0.168	0.158	0.166

Análisis de las funciones de mutación

Hemos propuesto varios esquemas para las mutaciones, debido a la importancia que teóricamente ésta operación posee en los GA. Así pues, planteamos 2 tasas de mutación (2% y 5%) considerando el caso de permitir o no que el mismo gen sea mutado en varias generaciones, en 4 funciones de mutación $m = \{m_1, m_2, m_3, m_4\}$. Todo ello planteado bajo el modelo factorial de efectos fijos (9.3), es decir, tenemos un diseño $2^2 \cdot 4$ lo que da a lugar a 16 posibles tipos de algoritmos y que analizamos a continuación mediante un ANOVA.

Los 16 algoritmos fueron procesados y codificados en lenguaje R para 15 generaciones. Cabe resaltar que todos los algoritmos convergen al valor máximo OPI luego de la primera generación, por tal razón hemos extraído el cromosoma que proporcione dicho óptimo, previniendo que se tratara de un óptimo local y así explorar más a fondo el comportamiento de dichos algoritmos. Entonces, se procesan nuevamente los 16 algoritmos en 15 generaciones. En este caso, todos los algoritmos encontraron el óptimo a partir de la quinta generación. A efectos de nuestro análisis hemos tomado los resultados de la quinta generación de cada algoritmo y realizado un

respectivo ANOVA, donde fue necesaria la transformación cuadrática de la respuesta OPI para el cumplimiento de los supuestos propios del análisis de la varianza. Los resultados se presentan en la tabla 9.4.

Tabla 9.4: ANOVA para las funciones de mutación. Caso 5.1

Efectos principales	SC	Gl	CM	F	P-valor
A: Reemp	0,0359505	1	0,0359505	2,20	0,1378
B: TasaM	0,302677	1	0,302677	18,54	0,0000
C: TipoM	0,861467	3	0,287156	17,59	0,0000
Interacciones					
AB	1,44034	1	1,44034	88,23	0,0000
AC	0,282535	3	0,0941782	5,77	0,0006
BC	0,986153	3	0,328718	20,14	0,0000
Residuos	101,46	6215	0,0163251		
Total	105,413	6227			

Como se puede ver en la tabla anterior, la tasa de mutación y el tipo de mutación resultan altamente significativos, siendo la condición de reemplazo no significativa en el análisis. Así mismo, se muestran como altamente significativas las interacciones. Por lo anterior se procede a la respectiva comparación de rangos múltiples, donde empleamos la prueba HSD de Tukey al 95 %. Dichas pruebas serán realizadas tanto a los efectos principales significantes como a aquellas interacciones donde dichos efectos estén involucrados.

En efecto, se realizan las respectivas pruebas de medias, resultando significativas en todos los casos. La figura 9.1 muestra la prueba de medias de Tukey para la tasa de mutación, donde se observan los mayores valores de OPI a una tasa de 2 %.

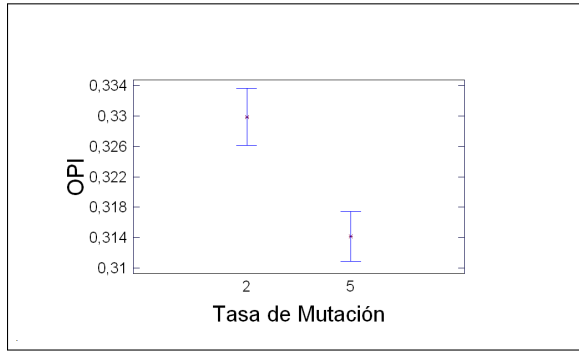


Figura 9.1: Prueba de Tukey para tasa de mutación. Caso 5.1

Por su parte, la figura 9.2 muestra la prueba de medias realizada al tipo de mutación donde los números 1, 2, 3, 4 representan las funciones de mutación $m_1(g)$, $m_2(g)$, $m_3(g)$ y $m_4(g)$ respectivamente. Entonces, se observa que es $m_1(g)$ quien aporta los mayores valores OPI.

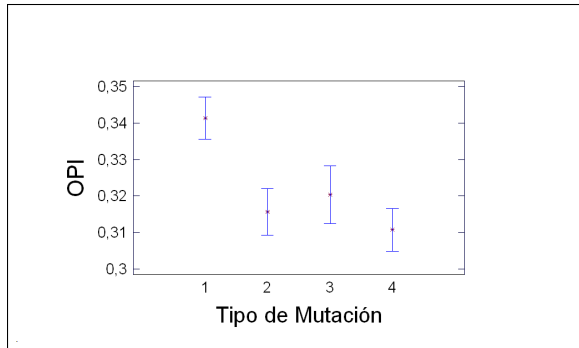


Figura 9.2: Prueba de Tukey para el tipo de mutación. Caso 5.1

Hasta ahora hemos visto el comportamiento de los factores significantes en el ANOVA por separado, pero a efecto de establecer un esquema general de mutación para nuestro GA y debido a la presencia de interacciones es necesario estudiar el comportamiento de dichos factores interactuando entre si. Se muestra la figura 9.3, donde se observa que los máximos OPI

permanecen con la función de mutación $m_1(g)$, pero en la tasa de mutación del 5%. Dicho esquema es el adoptado en el establecimiento de nuestro GA. En cuanto al factor de reemplazo o no del gen, hemos decidido permitir la mutación con reemplazo a fin de permitir a nuestro algoritmo una amplia búsqueda, diversificando plenamente los posibles valores por gen.

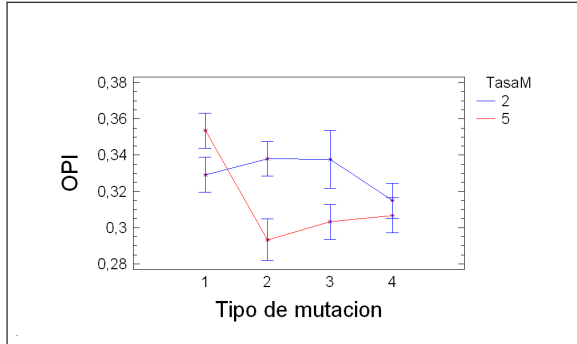


Figura 9.3: Gráfico de interacciones de efectos significantes. Caso 5.1

Predicción de nuevos cromosomas

La tabla 9.3 mostró los pesos asignados a cada gen, provenientes del algoritmo de modelización MRF_{P2} , donde resulta x_2 el gen con mayor peso, por lo tanto y en concordancia con todo lo expresado en la sección (9.4.1) y la ecuación (8.1), definimos la predicción de los nuevos cromosomas mediante la siguiente expresión.

$$\hat{y}_{h_i} = \left| \frac{\hat{y}_{m_i}(x_{2p_i} - x_{2h_i}) - \hat{y}_{p_i}(x_{2m_i} - x_{2h_i})}{x_{2p_i} - x_{2m_i}} \right| \quad (9.5)$$

donde \hat{y}_{m_i} y \hat{y}_{p_i} representan la predicción de la i -ésima madre y el i -ésimo padre provenientes de MRF_{P2} y x_{2p_i}, x_{2m_i} y x_{2h_i} corresponden a los valores del gen x_2 del i -ésimo padre, i -ésima madre e i -ésimo hijo.

Implementación del algoritmo genético al caso multivariante. Caso 5.1

Finalmente, se integran todos los componentes determinados anteriormente en nuestra propuesta de optimización, quedando el esquema de trabajo para el presente caso de estudio como se detalla a continuación.

Inicialización. Se crean aleatoriamente 400 cromosomas, formados por 8 genes: M_j , ($j = 1, 2, 3$), Z_l , ($l = 1, 2$) y X_k , ($k = 1, \dots, 6$) en escala normalizada.

- Hacer $z=b$
- Emplear MRF_{P2} para determinar los valores predictivos del vector $\hat{\mathbf{Y}} = (\hat{y}_1, \dots, \hat{y}_r)$.
- La asignación del peso de importancia a cada gen queda establecida por la tabla 9.3
- Llevar el conjunto de cromosomas con sus correspondientes predicciones a escala inicial. Así queda conformada la población inicial a_b , tal que a_{bi} representa el i -ésimo cromosoma de la población b .

Paso 1: Evaluación. Calcular los PMs correspondientes a cada tipo de respuesta del vector $\hat{\mathbf{Y}}$ y la medida OPI con la ecuación (9.4).

Paso 2: Selección. Calcular $a_{bi}^i = \{a_{bi}; OPI(a_{bi}) > \widehat{OPI}\}$.

- **Cruce:** Realizar cruce en a_{bi}^i .
- **Mutación:** Emplear función $m(1)$ (tabla 9.2) para la mutación de los genes, con tasa de mutación 5% y el factor S de reemplazo en todo el conjunto a_{bi}^i .

- **Predicción para la nueva generación:** $\forall a'_{bi}$ determinar el vector \hat{Y} de respuestas mediante la ecuación (9.5).

Paso 3: Reemplazo. Hacer $a_{Bi} = a_{bi} \cup a'_{bi}$.

Paso 4: Convergencia. Calcular $OPI_{a_{Bi}}$.

4.1 Si $\max[OPI_{a_{Bi}}] \leq OPI_{a_{bi}}$, hacer $z = b + 1$ y volver al paso 2.

En caso contrario, ir a 4.2.

4.2 ¿ $a_{z=B}$ converge?, en caso afirmativo ir al paso 5 y culminar.

Caso contrario, hacer $z = b + 1$ e ir al paso 2.

La figura 9.4 presenta los resultados de nuestra propuesta de optimización para 15 generaciones. Se observa una convergencia hacia el máximo valor OPI a partir de la generación 11, donde el máximo valor OPI se sitúa en 0.6731223, obteniendo como combinación óptima de los niveles de los parámetros (1.18, 2.77, 2.60, 1.64, 1.53, 1.33) con las respuestas estimadas $y_1=79.78$, $y_2= 9.99$ e $y_3= 1.64$.

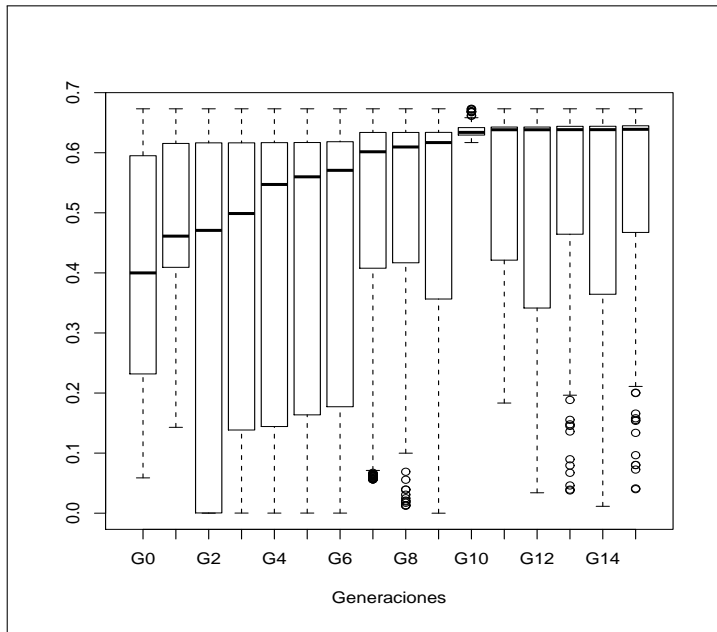


Figura 9.4: Comportamiento del GA multivariante en 15 generaciones. Caso 5.1

Chang (2008) en su trabajo muestra los resultados obtenidos en su propuesta en comparación con las 5 mejores combinaciones obtenidas en su anterior trabajo (Chang, 2006), donde la propuesta es basada en ANN. Entonces hemos decidido presentar dichos resultados y añadir los nuestros a fin de determinar numéricamente la validez de nuestra propuesta. Lo anterior se expresa en la tabla 9.5. Se puede notar que, al igual que la propuesta de Chang (2008), hemos podido explorar valores intermedios en los rangos de los factores de control, lo que en casos de aplicaciones reales podría significar un ahorro de recursos ó insumos. Así mismo hemos superado el valor OPI de Chang (2006) en gran cantidad y, en menos proporción, también hemos superado el valor OPI del mismo Chang (2008), lo que nos permite verificar la eficiencia de nuestra propuesta.

Tabla 9.5: Comparación de los resultados en diferentes propuestas de optimización multivariante. Caso 5.1

Método	Combinación de los parámetros						OPI
	x_1	x_2	x_3	x_4	x_5	x_6	
Nuestra propuesta	1.18	2.77	2.60	1.64	1.53	1.33	0.673122
Chang (2008)	1.98	2.98	1.06	2.44	2.96	1.06	0.668877
Chang (2006)	1	3	1	3	2	1	0.626443
	1	3	1	2	1	1	0.617955
	2	3	1	3	2	1	0.483897
	2	3	1	2	1	1	0.482461
	3	3	1	3	2	1	0.409331

9.3.2. Optimización del caso 5.2

Establecimiento de los componentes del AG

Población inicial y función Fitness

El presente caso de estudio corresponde a un diseño factorial de tipo 3^7 , el cual no se escapa de un gran número de combinaciones posibles de los niveles de los factores si lo consideramos fuera del arreglo ortogonal L_{18} . Por otra parte, el caso de estudio anterior nos mostró una clara convergencia al óptimo OPI en tan solo 15 generaciones, considerando una población inicial de 400 cromosomas, es decir, casi el 10% de todas las posibles combinaciones de los niveles de los parámetros en ese caso. Entonces, con la idea de explorar un poco más la eficiencia de nuestra propuesta de optimización y en vista de que se trata de una estructura más simple que el caso 5.1, hemos decidido llevar a cabo el análisis numérico del segundo caso de ilustración, considerando un número menor de cromosomas en la población inicial.

En efecto, decidimos generar aleatoriamente 200 cromosomas, (la mitad de

la generada en el caso 5.1) cuya estructura está conformada por 7 genes; 6 genes corresponden al vector de factores control $\mathbf{X} = \{A, B, C, D, E, F\}$ y 1 gen correspondiente al factor señal $M = \{0.1, 0.2, 0.3\}$. Dichos cromosomas estarán en escala normalizada a fin de calcular sus correspondientes respuestas con nuestro modelo predictivo $MRFP_2$. De esta forma, queda conformada la población inicial para la inicialización del algoritmo.

El objetivo del presente caso de estudio, análogamente al caso anterior, es la optimización simultánea del vector de respuestas $\mathbf{Y} = \{y_1, y_2, y_3\}$ bajo las características de calidad DLTB, DNTB y DSTB respectivamente, por lo que las funciones de deseabilidad definidas en la tabla 9.1 y la correspondiente medida de rendimiento OPI, ecuación (9.1), queda determinada por la expresión (9.6) y representará la función Fitness del algoritmo.

$$\begin{aligned}
 OPI = & \left\{ \exp \left[- \exp \left(- \frac{1}{3} \sum_{j=1}^3 \frac{\hat{y}_{1jk} - \hat{y}_{1j}^{\min}}{\hat{y}_{1j}^{\min}} \right) \right] \right. \\
 & \cdot \exp \left(- \frac{1}{3} \sum_{j=1}^3 \left| \frac{2\hat{y}_{2jk} - (y_{2j}^{\max} + y_{2j}^{\min})}{y_{2j}^{\max} - y_{2j}^{\min}} \right| \right) \\
 & \left. \cdot \exp \left[- \left(1 + \frac{1}{3} \sum_{j=1}^3 \frac{\hat{y}_{3jk} + y_{3j}^{\max}}{y_{3j}^{\max}} \right) \right] \right\}^{\frac{1}{3}} \quad (9.6)
 \end{aligned}$$

Como se puede notar, la diferencia entre las ecuaciones (9.4) y (9.6) radica en que el presente caso de estudio no posee factor de ruido, y por tanto, un sumatorio queda suprimido.

Cruce ponderado

La tabla 9.6 muestra las medidas de importancia provenientes del algoritmo MRF_{P2} ajustado al presente caso de estudio y acorde a la ecuación (8.2). Se observa a B como el gen con mayor importancia y por tanto con mayor peso, lo que influirá notoriamente en la formación de las correspondientes generaciones; además, marca el punto de interpolación en las predicciones de dichas generaciones como se detalla en el siguiente punto.

Tabla 9.6: Medidas de importancia y pesos de los genes según MRF_{P2} .
Caso 5.2

Gen	A	B	C	D	E	F
I_{xk}	0.006	0.011	0.008	0.009	0.006	0.009
Pesos	0.013	0.024	0.017	0.020	0.013	0.011

Predicción de nuevos cromosomas

La tabla 9.6 mostró los pesos asignados a cada gen, por tanto, la expresión (8.1) para la predicción de nuevos cromosomas queda establecida como lo muestra la ecuación (9.7), donde \hat{y}_{m_i} y \hat{y}_{p_i} representan la predicción de la i -ésima madre y el i -ésimo padre provenientes de MRF_{P2} y B_{p_i} , B_{m_i} y B_{h_i} corresponden a los valores del gen B del i -ésimo padre, i -ésima madre e i -ésimo hijo.

$$\hat{y}_{h_i} = \left| \frac{\hat{y}_{m_i} (B_{p_i} - B_{h_i}) - \hat{y}_{p_i} (B_{m_i} - B_{h_i})}{B_{p_i} - B_{m_i}} \right| \quad (9.7)$$

Esquema de mutación

A inicios del presente capítulo se expuso que este caso de estudio heredaba las componentes del GA. Entonces, es justamente este el punto a heredar, es decir, en base a los buenos resultados obtenidos en nuestra prueba piloto, implantamos nuestra propuesta de optimización mediante la función de mutación $m(1)$ (tabla 9.2), con tasa del 5% y permitiendo

el reemplazo en la mutación del gen. En el siguiente punto se presenta el procedimiento de nuestra propuesta de optimización al segundo caso de estudio multivariante.

Implementación del algoritmo genético al caso multivariante.

Caso 5.2

Inicialización. Se crean aleatoriamente 200 cromosomas, formados por 8 genes: M_j , ($j = 1, 2, 3$) y X_k , ($k = 1, \dots, 6$) en escala normalizada.

- Hacer $z=b$
- Emplear MRF_{P2} para determinan los valores predictivos del vector $\hat{\mathbf{Y}} = (\hat{y}_1, \dots, \hat{y}_r)$.
- La asignación del peso de importancia a cada gen queda establecido por la tabla 9.6
- Llevar el conjunto de cromosomas con sus correspondientes predicciones a escala inicial. Así queda conformada la población inicial a_b , tal que a_{bi} representa el i -ésimo cromosoma de la población b .

Paso 1: Evaluación. Calcular los PMs correspondientes a cada tipo de respuesta del vector $\hat{\mathbf{Y}}$ y la medida OPI con la ecuación (9.6).

Paso 2: Selección. Calcular $a'_{bi} = \{a_{bi} ; OPI(a_{bi}) > \widetilde{OPI}\}$.

- **Cruce:** Realizar cruce en a'_{bi} .
- **Mutación:** Emplear función $m(1)$ (tabla 9.2) para la mutación de los genes, con tasa de mutación 5 % y el factor S de reemplazo en todo el conjunto a'_{bi} .
- **Predicción para la nueva generación:** $\forall a'_{bi}$ determinar el vector $\hat{\mathbf{Y}}$ de respuestas mediante la ecuación (9.7).

Paso 3: Reemplazo. Hacer $a_{Bi} = a_{bi} \cup a'_{bi}$.

Paso 4: Convergencia. Calcular $OPI_{a_{Bi}}$.

4.1 Si $\max[OPI_{a_{Bi}}] \leq OPI_{a_{bi}}$, hacer $z = b + 1$ y volver al paso 2.
En caso contrario, ir a 4.2.

4.2 ¿ $a_{z=B}$ converge?, en caso afirmativo ir al paso 5 y culminar.
Caso contrario hacer $z = b + 1$ e ir al paso 2.

Paso 5: Obtener la combinación óptima de los niveles del factor control \mathbf{X} y su correspondiente multi-respuesta.

La figura 9.5 presenta los resultados del esquema anterior para 15 generaciones. Se observa una convergencia hacia el máximo valor OPI a partir de la generación 8, donde el máximo valor OPI se sitúa en 0.8840978. Entonces vemos cómo, a pesar de contar con una población inicial menor que en el caso 5.1 del presente capítulo, nuestro algoritmo logra encontrar el OPI óptimo sin mayor número de generaciones.

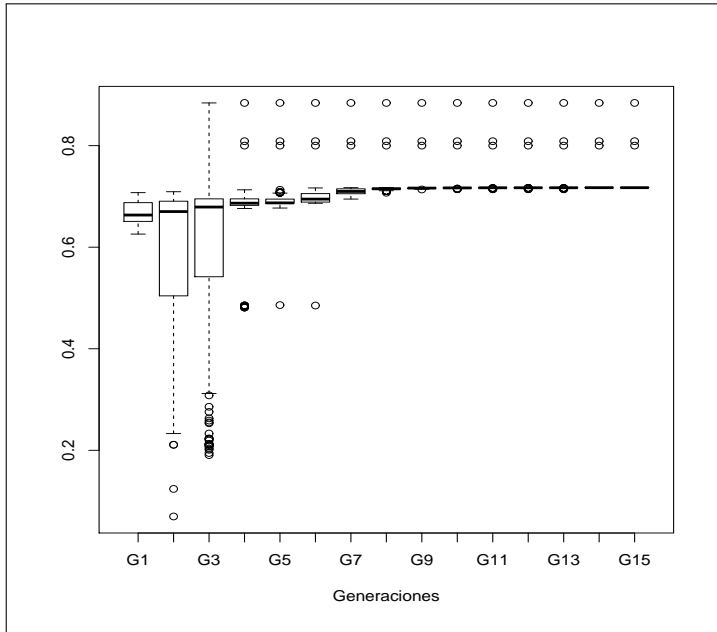


Figura 9.5: Comportamiento del GA multivariante en 15 generaciones. Caso 5.2

Chang y Chen (2011) en su trabajo presentan el valor OPI óptimo encontrado mediante su propuesta de optimización (basada en GA), en contraste con el máximo OPI hallado exclusivamente con las funciones PMs, bajo las 18 combinaciones establecidas en la matriz L_{18} . Resaltan las ventajas de su propuesta en cuanto a la maximización del valor OPI y a la posibilidad de encontrar valores continuos en el intervalo de los factores control. Ahora bien, nuestra propuesta de optimización no constituye un GA simple, sino que se nutre de las virtudes de nuestra propuesta de modelización, dando como resultado una búsqueda heurística más amplia (dentro y fuera de los rangos establecidos), la cual se enfatiza en los factores de control de mayor importancia del diseño, produciéndose así una notoria mejoría en el valor máximo OPI. Todo ello se refleja en la tabla 9.7.

En efecto, dicha tabla muestra los valores OPI encontrados y sus combinaciones en el vector de factores control \mathbf{X} ; nótese cómo entre nuestra propuesta y las restantes, las diferencias más profundas se encuentran, además de los valores OPI, en los niveles de los factores D y F, los cuales son los segundos factores más importantes y con mayor peso en nuestro esquema de modelización, los cuales fueron reflejados en la tabla 9.6.

Tabla 9.7: Comparación de los resultados en diferentes propuestas de optimización multivariante. Caso 5.2

Método	Combinación de los parámetros						OPI
	A	B	C	D	E	F	
Nuestra propuesta	1.66	3.06	2.42	2.21	1.48	2.86	0,8840978
Chang y Chen (2011)	1.85	2.99	2.97	1.00	1.86	1.00	0.753553
OPI mediante PMs	2	3	3	1	2	1	0.745636

Capítulo 10

Estrategias definitivas de optimización

El presente capítulo desarrolla las conclusiones de la última fase de nuestra propuesta metodológica, la cual tiene como objetivo la optimización en base a los modelos establecidos en la fase de modelización. Todo ello considerando problemas de tipo univariante y posteriormente generalizado al caso multivariante.

En el capítulo 7 mostramos las bases teóricas que dan lugar al diseño de las estrategias de optimización con el fin de evitar vacíos conceptuales. En el capítulo 8 se diseña el GA para el caso univariante, incorporando las propiedades dadas por PRED-RF en el establecimiento de los operadores genéticos en respuesta al objetivo específico 6. Seguidamente, en el capítulo 9, se generalizaron los resultados anteriores y se incorporaron las funciones de deseabilidad para el diseño de la estrategia de optimización en casos multivariantes, dando así respuesta al objetivo específico 7.

Así pues, nuestra propuesta de optimización consiste en un híbrido que combina las ventajas que ofrecen nuestros esquemas de modelización con los GA como metaheurística; todo ello desde el punto de vista univariante y posteriormente generalizado al caso multivariante. A continuación se comentan brevemente los aspectos más relevantes en cada caso y finalmente presentamos los esquemas de nuestra propuesta de optimización.

10.1. Casos univariantes

Nuestra propuesta de optimización para casos de tipo univariante consiste en un híbrido que combina las ventajas que ofrece PRED-RF en el reconocimiento de patrones y los GA como técnica de búsqueda heurística, donde el objetivo se concentra en optimizar la respuesta estimada de acuerdo a la característica de calidad establecida, bien sea de tipo dinámico o estático.

Definimos la metodología de optimización mediante el esquema que ofrecen los GA e incorporando las medidas de importancia de las variables proporcionada por PRED-RF en las operaciones de cruce. Adoptamos el cruce a 1 punto, pero hemos aumentado la probabilidad de cruce a los genes con mayor peso en PRED-RF. Todo ello quedó determinado por la función (8.2) y que mostramos nuevamente a continuación.

$$PC = \frac{I_{x_k}}{\sum_{k=1}^p I_{x_k}}$$

donde I_{x_k} representa la importancia del k -ésimo gen en PRED-RF, ($k = 1, \dots, p$).

Definimos la función de evaluación *fitness* como una expresión que permite la interpolación entre los nodos donde se ubiquen los nuevos individuos y

que se condiciona a los genes de mayor importancia en PRED-RF mediante la expresión:

$$\hat{y}_{h_i} = \left| \frac{\hat{y}_{m_i} (g_{p_i} - g_{h_i}) - \hat{y}_{p_i} (g_{m_i} - g_{h_i})}{g_{p_i} - g_{m_i}} \right|$$

donde \hat{y}_{h_i} representa la estimación de la respuesta del i -ésimo cromosoma hijo, \hat{y}_{m_i} y \hat{y}_{p_i} la predicción de la i -ésima madre y el i -ésimo padre provenientes de PRED-RF. Finalmente g_{p_i} , g_{m_i} y g_{h_i} corresponden a los valores del gen del i -ésimo padre, i -ésima madre e i -ésimo hijo, que serán determinados por las medidas de importancia de las variables proveniente de PRED-RF. Todo lo anterior fue detallado e ilustrado en el capítulo 8.

El proceso se inicia formando aleatoriamente n cromosomas. Los cromosomas están formados por $p = k + j + l$ genes, que corresponden a los k elementos del vector de factores de control, j niveles del factor señal y l niveles del factor ruido. La nueva generación estará formada por la unión de los cromosomas padres y los hijos con sus correspondientes valores respuesta \hat{y}_i . Como condición de parada se comparan las respuestas estimadas de la generación actual con la inicial o anterior en consideración a la característica de calidad bajo estudio, así como la convergencia de los cromosomas en un mismo rango de respuestas.

Numéricamente pudimos verificar las ventajas de nuestro esquema de optimización mediante 2 casos de ilustración provenientes de estudios reales. En primer lugar, comparamos nuestros resultados con los trabajos de Chiu y otros (1997) y Su y Chang (2000), permitiendo este ultimo compararnos con resultados provenientes de la propuesta Taguchi. Podemos decir que obtuvimos mejores resultados que los trabajos citados resaltando que, en

cuanto al trabajo más reciente (Su y Chang, 2000), obtuvimos una mejoría de más del 50 %, además de que se notan grandes discrepancias en cuanto a los niveles de los parámetros encontrados, lo que, a efectos de casos reales, podría traducirse en ahorros significativos de producción. El segundo caso de ilustración nos permitió ratificar el rendimiento de nuestra propuesta metodológica mediante un conjunto de datos con estructura más compleja; además, comparamos nuestros resultados con los obtenidos mediante las propuestas Taguchi, mostrado en Chang (2006), y el trabajo de Chang (2006).

Para finalizar la presente sección mostramos el esquema completo de nuestra propuesta de optimización al caso univariante mediante la figura 10.1.

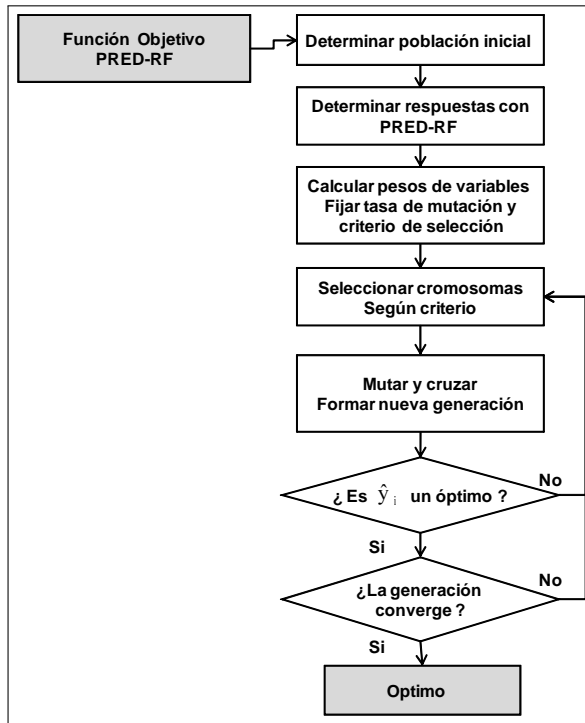


Figura 10.1: Propuesta de optimización para casos univariantes

10.2. Casos multivariantes

Nuestra propuesta de optimización a casos de tipo multivariante se presenta como una generalización del caso univariante, pero con algunos ingredientes extras producto del objetivo del manejo de medidas de calidad de forma simultánea. Nuestro híbrido ahora combina las propiedades de MRF, como nuestro esquema de modelización, las funciones de deseabilidad para el estudio de diferentes medidas de calidad simultáneamente y los GA como técnica de búsqueda heurística. Realizamos ajustes importantes en los componentes de los GA, comenzando por la función de evaluación, donde hemos incorporado la medida OPI como determinación del coste

de los cromosomas a las nuevas generaciones, cada d_j corresponde a los valores de deseabilidad en el j -ésimo vector de respuesta ($j = 1, \dots, r$) y el objetivo central de la estrategia de optimización se concentra en maximizar dicho valor OPI.

$$OPI = \left(\prod_{j=1}^r d_j \right)^{1/r}$$

Análogamente al caso univariante adoptamos el tipo de cruce de un punto, condicionando el punto de cruce a las medidas de importancia de las variables (genes) dadas por el MRF. Ello además dio lugar a la medida de ajuste de los nuevos cromosomas, la cual se determinó mediante interpolaciones en los nodos en la variable de mayor peso como se observó en (8.1) y que mostramos en la sección anterior.

Se propuso un esquema completo de mutación, donde se diseñaron mutaciones de tipo uniforme y no uniforme. Tal esquema quedó conformado por 16 posibles combinaciones entre tipos y tasas de mutación, las cuales se consideraban con o sin reemplazamiento del gen. Las funciones de mutación se mostraron en la tabla 9.2. Tal esquema de mutación fue estudiado mediante un ANOVA y empleando la prueba de medias de HSD Tukey al 95 % para los efectos significativos. Todo ello arrojó como resultado el establecimiento del esquema de mutación mediante una tasa del 5 %, permitiendo que el mismo gen sea mutado en diferentes generaciones y mediante la función $m_1(g)$, donde g representa el gen a mutar en el cromosoma correspondiente, g_{min} y g_{max} los valores mínimo y máximo del gen en la población.

$$m_1(g) = \begin{cases} g + g(0,10) & g < g_{max} \\ g & \text{en otros casos} \end{cases}$$

El proceso se inicia formando aleatoriamente n cromosomas. Los cromosomas están formados por $(2 + p)$ genes: el factor señal M_j , ($j = 1, \dots, s$), el factor de ruido Z_l , ($l = 1, \dots, n$) y el vector de k factores de control X_k , ($k = 1, \dots, p$) en escala normalizada.

Como criterio de selección en cada generación se emplea la mediana OPI, denotada como \widetilde{OPI} , y se seleccionan todos los cromosomas cuyos valores OPI sean mayores a \widetilde{OPI} . La nueva generación estará formada por la unión de los cromosomas padres y los hijos con su correspondiente vector \hat{Y} . Como condición de parada se comparan las respuestas estimadas de la generación actual con la inicial o anterior en consideración al OPI, así como y la convergencia de los cromosomas en un mismo rango de respuestas.

A fin de determinar las virtudes y deficiencias de nuestra propuesta se presentaron dos casos de ilustración. En el primer caso, empleado como prueba piloto, se realizaron comparaciones numéricas de los resultados obtenidos y los aportados en los trabajos de Chang (2006, 2008). Las conclusiones obtenidas en cuanto a las métricas empleadas y el esquema de mutación fueron transmitidas al segundo caso de ilustración, donde pudimos verificar la eficiencia de nuestra propuesta metodológica en contraste con el trabajo presentado por Chang y Chen (2011). En todos los casos, no sólo logramos mayores valores de OPI que los trabajos citados, sino que las discrepancias más notorias entre los valores de los parámetros determinados como óptimos se concentran en las variables cuyos pesos obtuvieron

los valores más altos de importancia en el MRF. De tal forma, validamos la eficiencia de nuestra propuesta de optimización y damos una virtud más al esquema de modelización MRF.

Para finalizar ésta sección se presenta la figura 10.2 en donde se resume el esquema nuestra propuesta de optimización para la mejora del diseño de parámetros en casos multivariantes.

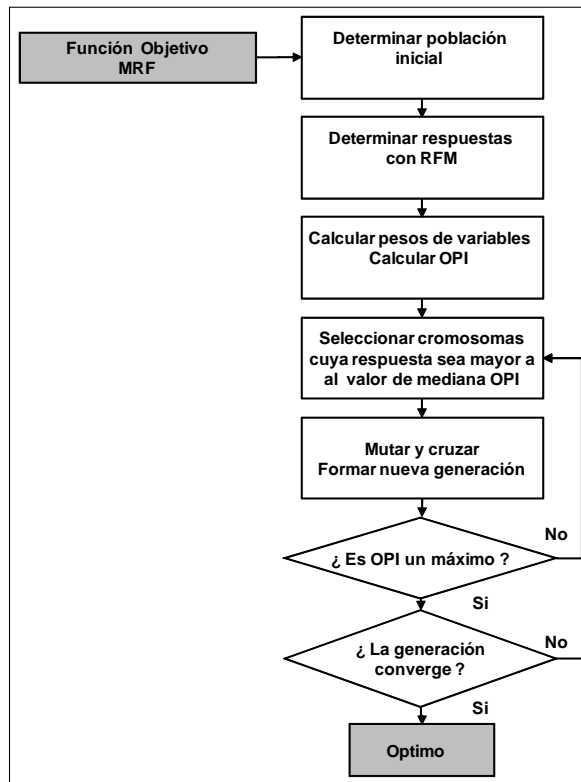


Figura 10.2: Propuesta de optimización para casos multivariantes

Parte IV

Conclusiones y futuras líneas de investigación

Capítulo 11

Conclusiones

Los capítulos 6 y 10 contienen los resultados y conclusiones, a nivel detallado, de las fases de modelización y optimización. No vamos a repetir aquí esa información sino que presentamos brevemente el cumplimiento de los objetivos y nuestra propuesta definitiva al problema del diseño de parámetros en respuesta a nuestro objetivo general. A la metodología propuesta la hemos llamado *métodos Forest-Genetic* y constituyen una alternativa a las últimas propuestas encontradas en la literatura, en problemas de tipo univariante y multivariante.

11.1. Cumplimiento de los objetivos

A continuación revisamos el logro de los objetivos específicos planteados, los cuales se numeran de acuerdo al orden establecido en el capítulo 1 del presente trabajo.

1. El capítulo 4 se inicia con la propuesta de 3 funciones de normalización con el fin de minimizar la variabilidad presente en

el conjunto de datos y unificar escalas con las ANN para las comparaciones pertinentes. Se determinó como criterio de selección de la mejor función de normalización la dispersión de los datos y el rendimiento general de los algoritmos, tal que los métodos predictivos que la incorporen presenten mayor robustez. Se selecciona la función f_2 presentada en la tabla 4.1.

2. En la sección 4.2 fueron diseñados y desarrollados los algoritmos PRED-T+VC, PRED-T+Boot, PRED-T+Bagg y PRED-RF.
3. En la sección 4.3 se presenta la estrategia basada en ANN. Seguidamente en la sección 4.4 y mediante 2 casos de ilustración diferentes se realizaron las comparaciones numéricas entre las estrategias mencionadas en el punto anterior. Considerando los criterios de robustez y estabilidad estadística se determinó al algoritmo PRED-RF como el esquema definitivo para la fase de modelización univariante bajo la mejor función de normalización, dada la importante mejora alcanzada en términos de RMSE en las distintas iteraciones ejecutadas.
4. Considerando los resultados anteriores e incorporando las métricas acordes, se diseñaron esquemas basadas en CART y RF como nuestra estrategia de modelización para casos multivariantes en el capítulo 5. A tal diseño lo llamamos MRF.
5. Seguidamente, en la sección 5.4 del capítulo 5, mediante 2 casos de ilustración diferentes se compararon los algoritmos MRF con las estrategias basadas en ANN, propuestas por otros autores. Todo ello condujo a la determinación de MRF como el mejor esquema para la modelización en términos de robustez y estabilidad estadística.
6. Bajo el esquema de GA se diseñó la estrategia de optimización

univariante en el capítulo 8. Se incorporaron las propiedades de PRED-RF en los operadores genéticos y se diseñó la función fitness como una medida de interpolación entre sus nodos. Además se optimizaron los dos casos de ilustración modelados en el capítulo 4 y se compararon los resultados finales con las propuestas metodológicas más recientes halladas en la literatura. Todo ello con el fin de establecer las virtudes y limitaciones del algoritmo propuesto.

7. En el capítulo 9 se diseñó el esquema de optimización al caso multivariante como una generalización del caso univariante. En este caso se incorporaron las propiedades del algoritmo MRF para el diseño de los operadores genéticos, se diseñó una función de interpolación entre sus nodos para la predicción de los nuevos cromosomas y las funciones de deseabilidad para el diseño de la medida OPI como función fitness. Seguidamente, y en correspondencia con el punto anterior, se optimizaron los dos casos de ilustración modelados en el capítulo 5 y se compararon los resultados finales con las propuestas metodológicas más recientes halladas en la literatura. Todo ello nuevamente con el fin de establecer las virtudes y limitaciones del algoritmo propuesto.
8. Numéricamente se constató que, tanto para el caso univariante como para el multivariante, nuestra propuesta metodológica se concentra en las variables de mayor importancia producto del proceso de modelización, lo que les permite alcanzar importantes mejoras en cuanto al objetivo de calidad considerado en cada caso.

11.2. Los métodos Forest-Genetic

Con lo anterior, se ha conseguido diseñar una metodología, a la que denominamos métodos Forest-Genetic, bajo las premisas de estabilidad

y robustez estadística. El empleo de funciones de normalización y RMSE como estadístico de prueba nos permitió minimizar la variabilidad presente en los modelos establecidos y cuidamos la estabilidad del proceso realizando diferentes números de iteraciones en las fases de modelización y optimización. Además, en la optimización se propusieron diferentes esquemas de mutación y se diseñaron los cruces en torno a los factores de control de mayor peso en el modelo.

La figura 11.1 ilustra el esquema completo de trabajo del método Forest-Genetic univariante donde hemos sombreado las 3 fases implicadas: normalización, modelización y optimización, tratadas en detalle en los capítulos 4 y 8.

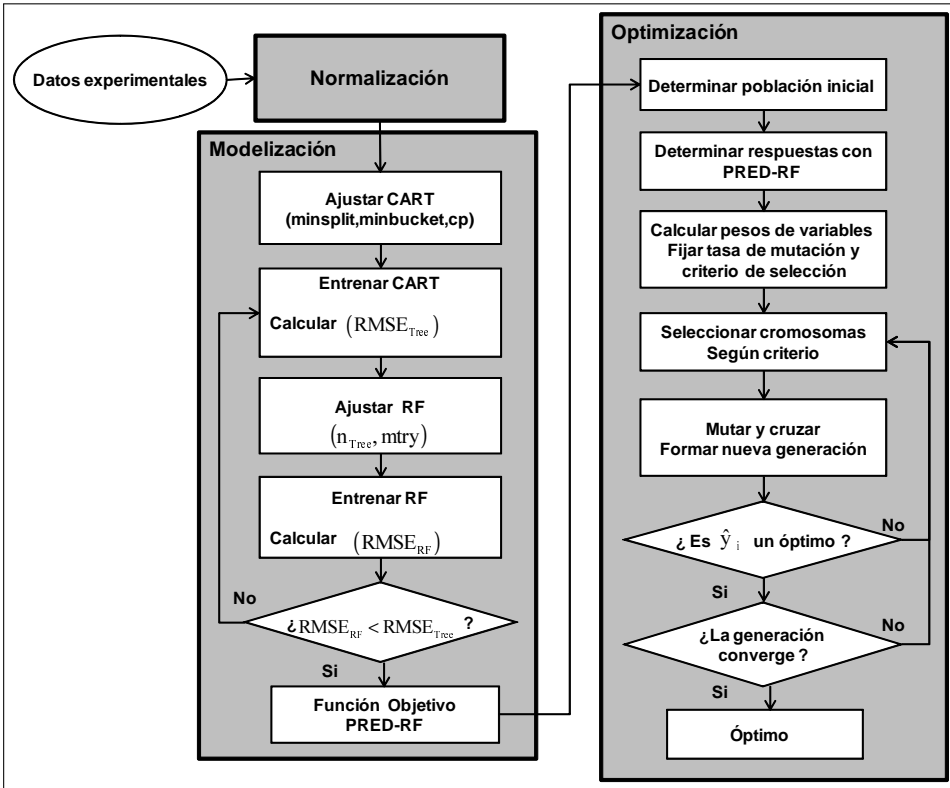


Figura 11.1: Método Forest-Genetic univariante

Análogamente, la figura 11.2 ilustra el esquema completo de Forest-Genetic multivariante, donde también hemos sombreado las 3 fases implicadas: normalización, modelización y optimización, tratadas en detalle en los capítulos 5 y 9.

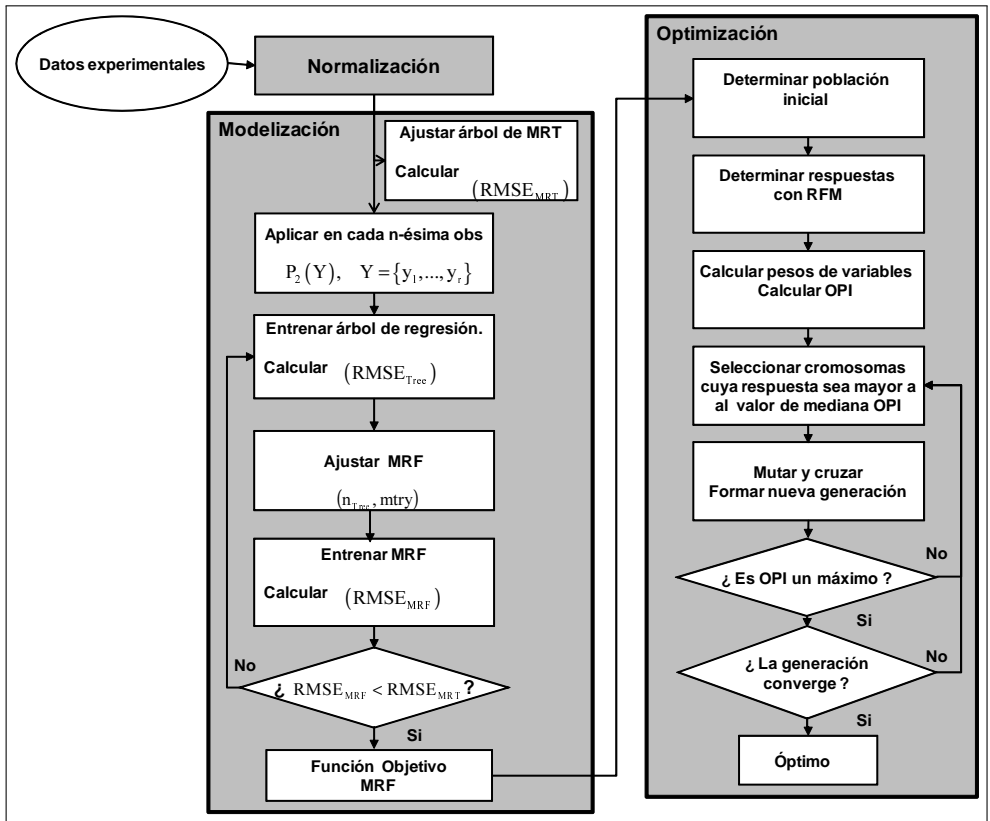


Figura 11.2: Método Forest-Genetic multivariante

Las principales virtudes de los métodos Forest-Genetic se resumen como sigue.

- Los métodos Forest-Genetic no presuponen independencia entre los factores ni linealidad de los factores con las respuestas, por tanto son perfectamente aplicables a casos de correlaciones entre factores y relaciones no lineales.
- Los métodos Forest-Genetic fueron diseñados considerando factores de ruido y señal con características de calidad de tipo dinámico

ó estático, por tanto pueden ser aplicados en casos de ausencias o presencia de dichos factores y en cualquier tipo de característica de calidad.

- Los métodos Forest-Genetic permiten optimizar la función de normalización, en virtud de la dispersion de los datos y el rendimiento general de los algoritmos.
- La fase de modelización de los métodos Forest-Genetic se basan en CART y RF, por tanto pueden ser usados en presencia de parámetros de tipo continuo y/o discreto, ampliando así su aplicación real.
- Los métodos Forest-Genetic en la fase de modelización permiten la monitorización gráfica de sus parámetros de control, lo que los hace más claros y accesibles en comparación a las ANN, que se presentan como una caja negra para el usuario.
- La fase de modelización de los métodos Forest-Genetic esta basada en CART y RF, lo que minimiza el riesgo de sobreajute de los modelos.
- El esquema de cruce diseñado en la fase de optimización permite a los métodos Forest-Genetic la dirección más eficiente en los hiperplanos de búsqueda.
- Los métodos Forest-Genetic permiten optimizar el esquema de mutación, lo cual fue probado numéricamente en los casos multivariantes, pero pueden ser perfectamente extendible al caso univariante.

Como se puede notar, todo lo anterior obedece a características teóricas que dotan a nuestros métodos de buenas propiedades. Sin embargo, vale la pena mencionar que los diferentes casos de ilustración tratados nos permitieron compararnos numéricamente con las últimas propuestas metodológicas halladas en la literatura. Verificamos que nuestra estrategia de

modelización se presenta más robusta y estable en presencia de diferentes números de iteraciones en comparación con las ANN. Asimismo, a lo largo del desarrollo de cada caso de ilustración observamos que los métodos Forest-Genetic requieren un número inferior de iteraciones tanto en la fase de modelización como en la de optimización para alcanzar buenos resultados en comparación con las metodologías que comprenden el uso de ANN y SA.

Investigaciones futuras

A fin de dar continuidad a la línea de *la optimización del diseño de parámetros*, se plantean las siguientes investigaciones a futuro:

- La robustez de los métodos Forest-Genetic en la fase de modelización fue estudiada considerando al RMSE como estadístico de prueba y se probaron diferentes números de iteraciones a fin de observar la estabilidad del proceso. En vista de las grandes diferencias encontradas en torno a este punto en contraste con otras metodologías se propone medir los tiempos de ejecución de los métodos Forest-Genetic mediante conjuntos de datos con estructura diferente.
- Introducir criterios económicos en la fase de optimización que tengan en cuenta el diferente impacto que cada característica tiene sobre los costos de calidad en la empresa.
- Utilizar los métodos Forest-Genetic en procesos de producción real a fin de contrastar la aplicabilidad práctica del mismo.
- Dado que los métodos Forest-Genetic han sido programados y ejecutados en lenguaje R y considerando las virtudes que tales

métodos proponen en la mejora del diseño de parámetros, resulta como extensión directa del presente trabajo el diseño del package Forest-Genetic como una contribución a la comunidad R. Asimismo se plantea el desarrollo de un software especializado y de fácil aplicabilidad en el entorno empresarial.

Bibliografía

- Antony, J. (2001). *Simultaneous optimization of multiple quality characteristics in manufacturing processes using Taguchi's quality loss function*. *Int. Adv Manuf Technol*17, 134–138.
- Bandyopadhyay, Sanghamitra y Sankar, Pal (2007). *Classification and learning using genetic algorithms. Applications in bioinformatics and web intelligence*. Springer. ISBN 13-978-3-540-49606-9.
- Box, George; Bisgaard, Soren y Fung, Conrad (1988). *An explanation and critique of taguchi's contributions to quality engineering. Quality and reliability engineering international*4, 123–131.
- Box, George E.P. y Meyer, Daniel R. (1986). *Dispersion effects from fractional designs. Technometrics*1, **28**, 19–27.
- Breiman, Leo (1996). *Bagging Predictors. Machine Learning*26, 123–140.
- Breiman, Leo (2001). *Random Fores. Machine Learning*45, 5–32.
- Breiman, Leo; Friedman, Jerome H.; Olsen, Richard A. y Stone, Charles J. (1984). *Classification and Regression Trees*. Chapman HALL/CRC.

-
- Breiman, Leo y Spector, Philip (1992). *Submodel selection and evaluation in regression: the X-random case*. *International Journal of Systems Science*3, 291–319.
- Castillo, Enrique; Montgomery, Douglas y McCarville, Daniel R. (1996). *Modified desirability functions for multiple response optimization*. *Journal of quality technology*28.
- Chang, Hsu-Hwa (2005). *Applications of neural networks and genetic algorithms to taguchi's robust design*. *International journal of electronic busines management*2, **3**, 90–96.
- Chang, Hsu-Hwa (2006). *Dynamic multi-response experiments by back-propagation networks and desirability functions*. *Journal of the chinese institute of industrial engineers*4, **23**, 280–288.
- Chang, Hsu-Hwa (2008). *A data mining approach to dynamic multiple responses in Taguchi experimental design*. *Expert Systems with Applications*35, 1095–1103.
- Chang, Hsu-Hwa y Chen, Yan-Kwang (2011). *Neuro-genetic approach to optimize parameter design of dynamic multiresponse experiments*. *Applied Soft Computing*11, 436–422.
- Chiu, Chih-Chou; Su, Chao-Ton; Yang, Gong-Hhung; Huang, Jeng-Sheng; Chen, Shia-Chung y Cheng, Nien-Tien (1997). *Selection of optimal parameter in gas-assisted injection moulding using a neural network model and the Taguchi method*. *International Journal of Quality Science*2, 106–120.
- De'ath, Glenn (2002). *Multivariate regression tree: a new technique for modeling species-environment relationships*. *Ecology*4, **83**, 1105–1117.

-
- De'ath, Glenn (2011). «Package mvpart». <http://cran.r-project.org/web/packages/mvpart/index.html>.
- Degertekin, S.O. (2007). *A comparison of simulated annealing and genetic algorithm for optimum design of nonlinear steel space frames. Struct. Multidisc. Optim*39, 347–359.
- Derringer, George y Suich, Ronald (1980). *Simultaneous optimization of several response variables. Journal of quality technology*4, **12**, 214–219.
- Dine, Abdessanad; Larocque, Denis y Bellavance, Francois (2009). *Multivariate trees for mixed outcomes. Computational statistics and data analysis*53, 3795–3804.
- Efron, Bradley y Tibshirani, Rob (1996). *Bootstrap Methods for Standard Error, Confidence Intervals and other measures of statistical accuracy. Statistical Science*1, **1**, 55–77.
- Harrington, Edwin C. (1965). *The desirability function. Industrial Quality Control*10, **21**, 494–498.
- Hastie, Trevor; Tibshirani, Robert y Friedman, Jerome (2009). *The elements of statistical learning. Data mining, inference and prediction. Springer series in statistic*.
- Haupt, Randy L. y Haupt, Sue Ellen (2004). *Practical Genetic Algorithms. John Wiley & Sons, Inc, second^a edición. ISBN 0-471-45565-2*.
- Hillier, Frederick S. y Lieberman, Gerald J. (2006). *Introducción a la investigación de operaciones. McGraw Hill / Interamericana editores, S.A., octava edición^a edición. ISBN 10:970-10-5621-3*.

-
- Hou, Tung-Hsu; Su, Chi-Hung y Chang, Hung-Zhi (2006). *Using neural networks and immune algorithms to find the optimal parameter for an IC wire bonding process. Expert Systems with Applications*34, 427–436.
- Hsieh, Kun-Lin; Tong, Lee-Ing; Chiu, Hung-Pin y Yeh, Hsin-Ya (2005). *Optimization of multi-response problem in Taguchi's dynamic system. Computers and Industrial Engineering*49, 556–571.
- Hsu, C.M.; Su, C.T. y Liao, D. (2004). *Simultaneous of the broadband tap coupler optical performance based on neural networks and exponential desirability functions. Int. Adv Manuf Technol*23.
- Hsu, Shang-Hsu; Lee, Feng-Liang y Wu, Muh-Cherng (2006). *An integrated approach to achieving optimal design of computer games. Expert Systems with Applications*31.
- Huang, M.L. y Hung, Y.H. (2006). *Combining radial basis function neural network and genetic algorithm to improve HDD driver IC chip scale package assembly yield. Expert Systems with Applications*1, **34**, 588–595.
- Jeyapaul, R.; Shahabudeen, P. y Krishnaiah (2006). *Simultaneous optimization of multi-response problems in the Taguchi method using genetic algorithm. Int. Adv Manuf Technol*30, 870–878.
- Joseph, V. Roshan y Wu, C.F Jeff (2002). *Robust parameter desing of multiple target systems. Technometrics*4, **44**, 338–346.
- Khuri, A.I y Conlom, M. (1981). *Simultaneous optimization of multiple responses represented by polynomial regression functions. Technometrics*23, **4**, 363–375.

-
- Kim, Kwang-Jae y Lin, Dennis K. (2000). *Simultaneous optimization of mechanical properties of steel by maximizing exponential desirability functions*. *Applied Statistics*3, **49**, 311–325.
- Kohabi, Ron (1995). «A study of Cross Validation and Bootstrap for Accuracy Estimation and Model Selection». En: *International Joint Conference on Artificial Intelligence (IJCAI)*, .
- Koksoy, Onur (2008). *A nonlinear programming solution to robust multi-response quality problem*. *Applied mathematics and computation*196, 603–612.
- Liaw, Andy y Matthew, Wiener (2009). «Package randomForest». <http://cran.r-project.org/web/packages/randomForest/index.html>.
- Logothetis, N. y Haigh, A. (1988). *Characterizing and optimizing multi-response processes by the Taguchi method*. *Quality and reliability engineering international*4, 159–169.
- Lucasius, C.B. y Kateman, G. (1993). *Understanding and using genetic algorithms. Part 1. Concepts, properties and context*. *Chemometrics and intelligent laboratory systems*19, 1–33.
- Maghsoodloo, Saed; Ozdemir, Gultekin; Jordan, Victoria y Huang, Chen-Hsiu (2004). *Strengths and limitations of Taguchi's contributions to quality, manufacturing and process engineering*. *Journal of Manufacturing Systems*2, **23**, 73–126.
- Michalewicz, Zbigniew (1996). *Genetic algorithms + Data structures = Evolutions Programs*. Artificial intelligence. Springer, third^a edición. ISBN 3-540-60676-9.

-
- Miller, Arden y Wu, C.F. Jeff (1996). *Parameter design for signal-response systems: A different look at Taguchi's dynamic parameter design*. *Statistical Science*2, **11**, 122–136.
- Montgomery, Douglas (2001). *Introduction to statistical quality control*. Wiley Sons, 4^a edición.
- Park, Yang-Gil y Yum, Bong-Jin (2003). *Development of performance measures for dynamic parameter design problems*. *International Journal of Manufacturing Technology and Management*5, 91–104.
- Peters, Andrea y Hothorn (2009). «Package ipred». <http://cran.uvigo.es/web/packages/ipred/index.html>.
- Qnet2000 (2000). «Neural network modeling for Windows». Vesta services inc.
- Robinson, Timothy J.; Borrór, Connie M. y Myers, Raymond (2004). *Robust parameter design: A review*. *Quality and reliability engineering international*20, 81–101.
- Rubio, Antonio; Altet, Josep; Aragonés, Xavier; González, Jose Luis; Mateo, Diego y Moll, Francesc (2000). *Diseño de circuitos y sistemas integrados*. Edicions de la Universitat Politècnica de Catalunya, SL, primera edición^a edición. ISBN 84-8301-404-1.
- Ryan, Thomas P (1989). *Statistical methods fro quality improvement*. Jhon Wiley & Sons. ISBN 0-471-84337-7.
- Segal, Mark R. (2003). «Machine Learning Benchmarks and Random Forest Regression». *Informe técnico*, Center for Bioinformatics & Molecular Biostatistics, University of California, San Francisco.

-
- Siroky, David S. (2005). *Navigating Random Forest and related advances in algorithmic modeling. Statistic Surveys*3, 147–163.
- Su, Chao-Ton y Chang, Hsu-Hwa (2000). *Optimization of parameter design: an intelligent approach using neural network and simulated annealing. International Journal of Systems Science*12, **31**, 1543–1549.
- Su, Chao-Ton; Chen, Mu-Chen y Chan, Hsiao-Ling (2005). *Applying neural network and scatter search to optimize parameter design with dynamic characteristics. Journal of the Operational Research Society*10, **56**, 1132–1140.
- Su, Chao-Ton y Tong, Lee-Ing (1997). *Multi-response robust design by principal component analysis. Total Quality Management*6, **9**, 409–416.
- Surajit, Pal y Susanta, Kumar (2010). *Multi-Response optimization using multiple regression based weighted signal to noise ratio (MRWSN). Quality Engineering*22, 336–350.
- Sutton, Clifton D. (2005). *Classification and regression tree, Bagging and Boosting. Handbook os Statistic*, 303–329.
- Tay, K.M. y Butler, C. (1997). *Modelling and optimizing of a mig welding process-A case study using experimental designs and neural networks. Quality and Reliability Engineering International*13, 61–70.
- Therneau, Terry M. y Atkinson, Beth (2010). «Package rpart». <http://cran.r-project.org/web/packages/rpart/index.html>.
- Tong, Lee-Ing; Su, Chao-Ton y Wang, Chung-Ho (1997). *The optimization of multi-response problems in the Taguchi method. International Journal of Quality & Reliability Management*4, **14**, 367–380.

-
- Tsenov, Aleksandar (2006). *Simulated Annealing and Genetic Algorithm in telecommunications network planning*. *International journal of computational intelligence*2, **4**, 240–245.
- Tsui, K-L (1996). *A critical look at Taguchi's modeling approach for robust design*. *Journal of Applied Statistics*1, **26**, 81–95.
- Wang, Chung-Ho y Tong, Lee-Ing (2005). *Optimization of dynamic multi-response problems using Grey multiple attribute decision making*. *Quality Engineering*17, 1–9.
- Wu, Ful-Chiang (2005). *Optimization of correlated multiple quality characteristics using desirability function*. *Quality engineering*17.
- Wu, Ful-Chiang y Yeh, Chi-Hao (2005). *Robust design of multiple dynamic quality characteristics*. *Int. Adv Manuf Technol*25, 579–588.
- Wu, Yun y Wu, Alan (1996). *Diseño robusto utilizando los métodos Taguchi*. Editorial Díaz de Santos. ISBN 84-7978-305-2.
- Zang, M.I., C.and Friswell y Mottershead, J.E. (2005). *A review of robust optimal design and its application in dynamics*. *Computer and Structures*83, 315–326.

A. Datos casos 4.1 y 4.2

Caso de ilustración 4.1

Tabla 12.1: Conjunto de datos. Caso 4.1

No.	Factores de control								Respuestas				
	A	B	C	D	E	F	G	H	y_1	y_2	y_3	y_4	y_5
1	50	230	50	1	90	64	0	0	42	40	57	68	74
2	50	230	60	1.5	110	65	0.5	3	71	76	74	74	75
3	50	230	70	2	130	66	1	6	84	80	83	80	82
4	50	240	50	1	110	65	1	6	37	29	34	38	41
5	50	240	60	1.5	130	66	0	0	117	115	121	123	116
6	50	240	70	2	90	64	0.5	3	37	36	36	39	36
7	50	250	50	1.5	90	66	0.5	6	85	87	88	93	90
8	50	250	60	2	110	64	1	0	28	26	24	25	29
9	50	250	70	1	130	65	0	3	84	79	84	79	73
10	60	230	50	2	130	65	0.5	0	74	84	64	69	65
11	60	230	60	1	90	66	1	3	84	87	95	88	94
12	60	230	70	1.5	110	64	0	6	71	68	68	70	65
13	60	240	50	1.5	130	64	1	3	25	24	25	28	24
14	60	240	60	2	90	65	0	6	88	88	89	90	79
15	60	240	70	1	110	66	0.5	0	114	124	125	117	118
16	60	250	50	2	110	66	0	3	106	106	104	99	107
17	60	250	60	1	130	64	0.5	6	31	41	43	36	40
18	60	250	70	1.5	90	65	1	0	60	53	58	51	60

Caso de ilustración 4.2

Tabla 12.2: Conjunto de datos. Caso 4.2

No.	Factores de control						Factor ruido	Respuestas				
	A	B	C	D	E	F						
1	0.32	1	0.2	40	0.01	100	N_1	0.082	0.084	0.085	0.076	0.079
							N_2	0.125	0.127	0.124	0.122	0.155
2	0.032	2	0.035	50	0.016	125	N_1	0.106	0.108	0.105	0.110	0.108
							N_2	0.148	0.148	0.154	0.155	0.152
3	0.032	3	0.05	60	0.02	150	N_1	0.157	0.184	0.187	0.184	0.169
							N_2	0.157	0.184	0.187	0.184	0.169
4	0.036	2	0.035	40	0.016	150	N_1	0.177	0.184	0.181	0.191	0.194
							N_2	0.193	0.211	0.2	0.195	0.2
5	0.36	3	0.05	50	0.02	100	N_1	0.095	0.092	0.090	0.092	0.093
							N_2	0.157	0.155	0.124	0.150	0.153
6	0.036	4	0.065	30	0.01	125	N_1	0.001	0.001	0.001	0.001	0.001
							N_2	0.2	0.2	0.2	0.2	0.2
7	0.043	3	0.065	20	0.02	125	N_1	0.086	0.081	0.086	0.085	0.085
							N_2	0.129	0.130	0.135	0.118	0.118
8	0.043	4	0.08	30	0.01	150	N_1	0.001	0.001	0.001	0.001	0.001
							N_2	0.2	0.2	0.2	0.2	0.2
9	0.043	5	0.05	40	0.016	100	N_1	0.267	0.267	0.275	0.275	0.263
							N_2	0.274	0.263	0.260	0.246	0.260
10	0.032	1	0.05	60	0.016	125	N_1	0.347	0.358	0.262	0.296	0.282
							N_2	0.285	0.299	0.235	0.246	0.245
11	0.032	2	0.02	40	0.02	150	N_1	0.181	0.178	0.168	0.171	0.189
							N_2	0.214	0.213	0.209	0.201	0.206
12	0.032	3	0.035	50	0.01	100	N_1	0.303	0.292	0.291	0.288	0.293
							N_2	0.239	0.221	0.229	0.235	0.235
13	0.036	2	0.05	50	0.01	150	N_1	0.128	0.143	0.144	0.135	0.131
							N_2	0.170	0.176	0.161	0.173	0.163
14	0.036	3	0.65	30	0.016	100	N_1	0.001	0.001	0.001	0.001	0.001
							N_2	0.2	0.2	0.2	0.2	0.2
15	0.036	4	0.035	40	0.02	125	N_1	0.074	0.073	0.074	0.066	0.072
							N_2	0.088	0.081	0.138	0.142	0.152
16	0.043	3	0.08	30	0.02	100	N_1	0.076	0.088	0.099	0.095	0.090
							N_2	0.208	0.157	0.156	0.177	0.161
17	0.043	4	0.05	40	0.01	125	N_1	0.305	0.308	0.302	0.298	0.292
							N_2	0.257	0.256	0.258	0.257	0.244
18	0.043	5	0.065	20	0.016	150	N_1	0.001	0.001	0.001	0.001	0.001
							N_2	0.2	0.2	0.2	0.2	0.2

B. Datos casos 5.1 y 5.2

Caso de ilustración 5.1

Tabla 12.3: Conjunto de datos. Caso 5.1

No.	Factor ruido	Respuestas					
		y_1					
		$M_1 = 10$		$M_2 = 20$		$M_3 = 30$	
1	$N1$	61.6	78.2	128.0	106.0	230.6	226.9
	$N2$	70.8	57.1	137.3	160.3	282.2	252.5
2	$N1$	88.3	93.6	175.2	181.5	259.7	304.5
	$N2$	72.9	72.7	174.0	145.5	258.4	214.6
3	$N1$	80.8	81.1	154.3	157.4	238.1	237.8
	$N2$	77.2	83.3	167.1	159.0	251.8	257.9
4	$N1$	65.9	71.3	179.2	151.5	196.1	221.6
	$N2$	83.7	78.4	135.6	177.0	246.9	291.8
5	$N1$	79.4	88.6	121.9	151.6	248.8	245.1
	$N2$	67.8	87.3	113.6	141.3	171.5	244.7
6	$N1$	90.5	87.0	161.8	169.4	286.9	236.5

Continúa en la página siguiente

		continuación					
7	N2	87.6	87.8	160.7	163.9	231.4	288.7
	N1	80.9	74.7	165.9	163.4	232.2	246.4
8	N2	69.9	78.7	141.7	159.1	260.4	239.7
	N1	92.3	71.7	185.7	154.3	233.1	240.5
9	N2	104.8	89.4	216.1	173.0	340.6	308.9
	N1	92.8	59.8	130.7	142.2	257.3	266.0
10	N2	82.1	87.0	175.1	138.2	161.4	274.4
	N1	86.0	100.0	179.1	175.4	246.9	244.9
11	N2	81.9	91.2	190.8	137.8	293.5	227.1
	N1	76.3	78.2	140.0	154.7	264.1	260.1
12	N2	67.1	76.0	169.5	175.4	239.2	251.8
	N1	91.4	81.8	160.1	167.2	238.5	197.2
13	N2	85.1	63.8	123.8	166.4	233.5	242.0
	N1	87.9	82.5	146.5	167.0	212.4	222.2
14	N2	57.4	78.5	91.6	182.3	250.3	207.3
	N1	88.1	78.1	156.7	170.1	239.1	215.2
15	N2	81.7	75.7	140.2	127.8	241.4	211.7
	N1	101.8	78.2	168.3	180.8	240.6	235.2
16	N2	80.4	76.5	206.7	222.6	325.1	285.2
	N1	77.4	75.4	171.7	159.0	201.3	219.7
17	N2	72.0	69.5	189.1	168.6	254.3	237.3
	N1	71.4	69.2	145.0	152.5	223.8	218.7
18	N2	77.0	70.5	158.4	154.0	218.4	224.1
	N1	82.8	67.8	183.7	175.5	276.1	254.4
	N2	85.2	92.0	154.4	157.6	249.3	286.1
		y_2					
		$M_1 = 10$		$M_2 = 20$		$M_3 = 30$	

Continúa en la página siguiente

continuación

1	<i>N1</i>	7.4	7.2	16.7	13.2	23.7	24.1
	<i>N2</i>	9.1	10.2	22.8	17.8	26.2	26.7
2	<i>N1</i>	10.1	8.8	23.4	22.6	29.6	30.3
	<i>N2</i>	8.7	9.2	19.1	24.1	31.4	30.5
3	<i>N1</i>	10.8	10.5	20.1	20.9	30.6	32.4
	<i>N2</i>	10.6	10.9	21.8	23.7	30.5	32.7
4	<i>N1</i>	7.6	7.2	15.3	14.8	22.5	22.2
	<i>N2</i>	8.1	7.6	14.1	14.7	21.9	21.8
5	<i>N1</i>	11.9	12.7	25.6	25.7	36.6	35.7
	<i>N2</i>	10.5	11.8	25.8	26.2	39.1	33.1
6	<i>N1</i>	10.2	10.2	23.7	21.8	32.0	32.8
	<i>N2</i>	10.7	11.2	22.5	20.0	34.2	28.0
7	<i>N1</i>	11.7	12.1	23.3	23.3	33.5	32.7
	<i>N2</i>	11.6	11.6	22.0	22.5	33.8	34.0
8	<i>N1</i>	8.4	16.9	18.5	28.1	27.3	2.8
	<i>N2</i>	8.2	6.1	18.5	15.1	29.1	21.1
9	<i>N1</i>	8.6	9.7	17.3	18.4	30.9	31.5
	<i>N2</i>	8.2	8.4	18.6	18.0	31.3	30.9
10	<i>N1</i>	6.9	6.6	15.8	16.4	22.2	24.9
	<i>N2</i>	8.3	7.2	14.3	16.6	23.1	24.2
11	<i>N1</i>	9.5	9.2	27.9	25.4	32.3	33.8
	<i>N2</i>	10.8	12.2	23.6	20.4	26.8	29.2
12	<i>N1</i>	11.8	10.5	22.9	25.9	36.9	36.8
	<i>N2</i>	11.6	11.5	22.1	22.6	32.5	31.9
13	<i>N1</i>	10.1	10.3	19.9	22.2	27.1	27.7
	<i>N2</i>	10.6	10.1	20.1	19.2	27.8	24.6
14	<i>N1</i>	12.1	10.0	24.1	20.2	28.0	35.6

Continúa en la página siguiente

								continuación
15	<i>N2</i>	11.3	11.8	23.3	24.4	32.1	38.9	
	<i>N1</i>	10.2	7.6	14.9	19.4	26.9	19.5	
16	<i>N2</i>	8.7	7.2	16.8	14.1	26.6	25.1	
	<i>N1</i>	10.4	10.6	20.7	22.2	34.4	30.0	
17	<i>N2</i>	11.1	10.9	20.6	21.4	30.5	31.4	
	<i>N1</i>	8.8	8.4	19.0	13.8	26.2	24.1	
18	<i>N2</i>	9.2	9.0	16.7	17.4	27.1	26.2	
	<i>N1</i>	10.7	9.2	19.8	20.1	27.3	31.0	
	<i>N2</i>	11.3	7.7	19.3	22.6	29.4	26.6	
		y_3						
		$M_1 = 10$			$M_2 = 20$			$M_3 = 30$
1	<i>N1</i>	1.9	1.9	4.6	3.7	7.6	4.6	
	<i>N2</i>	2.0	2.1	3.9	4.8	4.7	4.3	
2	<i>N1</i>	1.8	2.0	4.0	2.8	6.0	3.3	
	<i>N2</i>	1.9	2.2	3.8	3.5	6.7	6.1	
3	<i>N1</i>	1.0	3.2	4.4	5.4	8.1	6.7	
	<i>N2</i>	1.4	2.6	3.8	3.9	8.0	2.7	
4	<i>N1</i>	1.7	2.2	4.1	4.9	5.9	6.0	
	<i>N2</i>	2.1	2.6	3.6	3.3	7.7	6.9	
5	<i>N1</i>	2.0	2.1	2.7	3.8	4.8	5.8	
	<i>N2</i>	2.6	1.7	4.1	3.6	5.5	5.2	
6	<i>N1</i>	1.8	2.2	2.5	4.1	4.9	5.4	
	<i>N2</i>	1.2	2.0	4.0	4.4	6.6	3.1	
7	<i>N1</i>	1.9	1.6	5.2	5.1	6.6	5.3	
	<i>N2</i>	1.9	1.7	3.6	5.1	5.8	5.1	
8	<i>N1</i>	2.8	1.5	3.4	3.4	6.2	3.7	
	<i>N2</i>	1.8	2.2	4.1	4.3	9.0	5.0	

Continua en la página siguiente

continuación

9	<i>N1</i>	2.3	1.7	4.5	5.1	7.4	6.5
	<i>N2</i>	2.2	1.2	4.0	4.0	7.2	6.7
10	<i>N1</i>	1.7	1.7	3.9	5.8	5.2	8.4
	<i>N2</i>	2.1	1.8	3.9	4.8	4.5	8.0
11	<i>N1</i>	1.0	1.7	4.1	4.0	3.9	5.6
	<i>N2</i>	2.1	2.3	5.0	4.6	6.7	4.5
12	<i>N1</i>	2.1	1.9	4.0	3.7	7.4	6.2
	<i>N2</i>	1.6	0.9	4.2	4.1	2.1	6.9
13	<i>N1</i>	2.0	2.3	5.0	4.7	6.7	7.3
	<i>N2</i>	2.2	2.1	4.5	4.7	7.4	6.2
14	<i>N1</i>	1.7	2.1	3.8	3.3	7.2	4.9
	<i>N2</i>	1.8	2.7	3.1	3.8	4.9	4.7
15	<i>N1</i>	1.7	2.2	5.3	3.6	5.0	7.5
	<i>N2</i>	2.2	1.9	4.8	2.6	3.8	5.5
16	<i>N1</i>	1.9	2.3	3.1	4.2	5.8	3.3
	<i>N2</i>	2.1	1.6	4.1	4.8	5.4	6.9
17	<i>N1</i>	1.5	2.1	3.8	4.4	6.9	4.2
	<i>N2</i>	1.6	1.8	3.7	4.4	4.6	4.2
18	<i>N1</i>	2.5	2.2	3.1	3.4	7.7	7.4
	<i>N2</i>	1.6	1.6	3.9	4.7	6.0	7.3

Caso de ilustración 5.2

Tabla 12.4: Conjunto de datos. Caso 5.2

N°	Respuestas					
	y_1					
	$M_1 = 10$		$M_2 = 20$		$M_3 = 30$	
1	7.80	8.13	14.22	14.92	25.96	28.84
2	8.63	7.53	17.01	16.52	27.13	31.76
3	8.12	7.28	16.65	15.84	25.98	26.05
4	8.18	8.07	18.29	15.92	25.34	20.76
5	7.04	7.58	13.11	16.53	27.66	22.89
6	8.32	9.79	16.80	14.74	26.55	26.82
7	8.02	8.30	14.46	15.42	25.74	23.10
8	6.36	8.24	18.23	17.48	20.24	28.28
9	5.93	8.65	16.51	13.43	22.36	19.92
10	8.56	8.88	17.57	19.17	25.73	23.20
11	7.61	9.85	17.34	16.31	27.06	28.60
12	7.88	8.07	16.89	12.55	22.98	24.26
13	8.73	6.82	18.22	15.64	25.64	20.26
14	7.97	9.72	16.72	11.98	23.27	23.10
15	9.16	8.77	16.72	15.86	24.97	30.30
16	9.32	8.71	14.86	15.67	21.87	28.43
17	8.32	6.91	16.03	14.10	22.70	18.87
18	8.71	6.37	14.87	18.74	31.61	22.69
1	y_2					
	$M_1 = 10$		$M_2 = 20$		$M_3 = 30$	
	0.98	1.09	1.63	1.42	2.79	5.53

Continua en la página siguiente

continuación

2	1.02	1.05	2.22	1.73	3.14	3.44
3	1.05	0.94	2.17	2.15	2.90	2.92
4	0.68	0.72	1.46	1.50	2.19	2.26
5	1.14	1.23	2.64	2.27	3.44	3.98
6	1.00	0.96	2.49	1.97	3.36	2.95
7	1.22	1.20	2.29	2.39	3.18	3.29
8	0.73	0.86	1.43	2.13	2.11	2.18
9	1.12	0.91	1.92	1.77	2.52	2.98
10	0.80	0.75	1.45	1.62	2.36	2.40
11	0.92	1.23	2.55	2.54	3.95	3.47
12	1.08	1.05	2.28	2.22	3.32	3.23
13	0.95	0.99	2.00	2.00	2.94	2.93
14	1.17	1.14	1.95	2.35	3.91	3.58
15	0.85	0.79	1.42	1.75	2.33	2.34
16	1.05	1.10	2.01	2.26	3.29	3.02
17	0.80	0.85	2.07	1.99	2.71	2.46
18	1.14	0.98	1.92	1.58	3.57	2.97
	y_3					
	$M_1 = 10$		$M_2 = 20$		$M_3 = 30$	
1	15.00	16.53	32.24	41.81	48.25	83.92
2	16.00	16.30	42.92	29.17	57.08	41.51
3	23.00	23.29	40.63	48.37	34.52	57.33
4	25.00	15.98	37.74	41.75	59.79	47.14
5	18.00	14.40	23.80	44.36	41.62	43.45
6	26.00	10.28	40.45	30.69	23.84	67.64
7	28.00	19.68	40.57	50.66	61.05	72.99
8	12.00	26.70	31.01	32.74	82.76	66.55

Continúa en la página siguiente

continuación

9	17.00	19.78	49.92	28.39	56.18	52.64
10	21.00	28.16	39.08	47.59	71.62	83.71
11	20.00	16.24	43.19	28.68	60.13	70.66
12	18.00	11.38	46.14	22.51	66.97	65.73
13	26.00	22.32	64.67	40.40	94.98	58.26
14	16.00	23.16	24.82	44.13	51.38	63.52
15	14.00	12.88	40.57	33.27	33.99	60.82
16	22.00	15.90	51.58	43.90	75.55	86.55
17	23.00	20.34	42.91	32.95	36.92	64.79
18	19.00	12.43	37.70	38.89	69.16	55.98