

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
DEPARTAMENTO DE INGENIERÍA MECÁNICA



TESIS DOCTORAL

ALGORITMOS EVOLUTIVOS Y PREFERENCIAS DEL
DECISOR APLICADOS A PROBLEMAS DE OPTIMIZACIÓN
MULTIOBJETIVO DISCRETOS

Máximo Méndez Babey

Septiembre 2008

ANEXO II
UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
DEPARTAMENTO DE INGENIERÍA MECÁNICA



PROGRAMA DE DOCTORADO:
TECNOLOGÍA INDUSTRIAL

TESIS DOCTORAL

ALGORITMOS EVOLUTIVOS Y PREFERENCIAS DEL
DECISOR APLICADOS A PROBLEMAS DE OPTIMIZACIÓN
MULTIOBJETIVO DISCRETOS

El Director

El Codirector

El Autor

Blas Galván González David Greiner Sánchez Máximo Méndez Babey

Las Palmas de Gran Canaria, Septiembre de 2008

Índice General

Índice de Tablas	vii
Índice de Figuras	xiv
Lista de Acrónimos	xv
Agradecimientos	xvii
Introducción general	1
Presentación	1
Resultados y contribuciones	3
Publicaciones realizadas	5
Organización del documento	5
1 Ayuda Multicriterio a la Decisión	7
1.1 Paradigmas Unicriterio y Multicriterio	7
1.2 Problema de decisión	8
1.3 Actores que intervienen en un proceso de decisión	9
1.4 Elaboración del modelo de decisión	11
1.5 Las acciones	11
1.6 Las diferentes problemáticas	12
1.7 Modelado de las preferencias	12
1.7.1 Relación binaria	13
1.7.2 Sistema relacional de preferencias	15
1.7.3 Estructuras de preferencias clásicas	16
1.8 Modelado de las consecuencias	21
1.9 Los criterios	23
1.10 La agregación	23
1.10.1 Métodos de agregación completa	24
1.10.2 Métodos de agregación parcial	32
2 Optimización Multiobjetivo	39
2.1 Multiobjetivo-Multicriterio: breve discusión	39
2.2 Clasificación de los métodos de Optimización Multiobjetivo	39

2.3	Conceptos básicos	41
2.3.1	Problema Multiobjetivo	41
2.3.2	Dominancia	42
2.3.3	Relaciones derivadas de la dominancia.	44
2.3.4	Puntos de referencia	45
2.4	Métodos que articulan preferencias a posteriori	46
2.4.1	Métodos de los pesos	47
2.4.2	Método de la distancia a un punto de referencia	50
2.4.3	Método de las ϵ -restricciones	53
2.5	Métodos que articulan preferencias a priori	55
2.5.1	Programación por metas	56
2.5.2	Ordenamiento lexicográfico	59
3	Algoritmos Evolutivos	61
3.1	Introducción	61
3.2	De los modelos biológicos a los programas informáticos	62
3.3	Paradigmas de los Algoritmos Evolutivos.	63
3.4	Terminología.	66
3.5	Codificación de variables	69
3.6	Operadores evolutivos básicos	71
3.6.1	Operadores de selección.	71
3.6.2	Operadores de cruce.	73
3.6.3	Operadores de mutación.	74
3.7	Otros operadores evolutivos.	75
3.8	Equilibrio Explotación-Exploración	79
4	Algoritmos Evolutivos Multiobjetivo. Estado del Arte	81
4.1	Introducción	81
4.2	Clasificación de los AEMOs	84
4.3	AEMOs que asumen preferencias a posteriori	89
4.3.1	RWGA	90
4.3.2	VEGA	91
4.3.3	MOGA	92
4.3.4	NSGA - NSGAI	94
4.3.5	ϵ -MOEA	95
4.4	AEMOs que introducen preferencias a priori	97
4.4.1	R-NSGAI	102
4.4.2	RD-NSGAI	103
4.5	Evaluación de los resultados	104
4.5.1	La métrica C	105
4.5.2	La métrica S	106
4.5.3	ϵ -Indicadores	107
4.6	Problema test	108

4.6.1	Problema test ZDT5 de Deb	109
5	Contribuciones	111
5.1	Introducción	111
5.2	TOPSISGA	112
5.2.1	Introducción	112
5.2.2	Algoritmo propuesto	113
5.2.3	Resultados experimentales	113
5.2.4	Conclusiones	117
5.3	WP-TOPSISGA	117
5.3.1	Introducción	117
5.3.2	Algoritmo propuesto	118
5.3.3	Resultados experimentales	119
5.3.4	Conclusiones	120
5.4	CMOEA&TOPSIS	124
5.4.1	Introducción	124
5.4.2	Algoritmo propuesto	125
5.4.3	Resultados experimentales	125
5.4.4	Conclusiones	131
5.5	CMOEA20-80&TOPSIS	131
5.5.1	Introducción	131
5.5.2	Algoritmo propuesto	132
5.5.3	Resultados experimentales	133
5.5.4	Conclusiones	135
6	Aplicaciones a dos Problemas Multiobjetivo Discretos	137
6.1	Introducción	137
6.2	Problema de la Mochila Multiobjetivo	138
6.2.1	Descripción del problema	138
6.2.2	Datos adoptados	140
6.2.3	Configuración de parámetros	140
6.2.4	Métricas	140
6.2.5	Análisis de resultados con TOPSISGA	140
6.2.6	Análisis de resultados con WP-TOPSISGA	142
6.2.7	Análisis de resultados con CMOEA&TOPSIS	147
6.2.8	Análisis de resultados con CMOEA20-80&TOPSIS	152
6.2.9	Conclusiones	152
6.3	Problema de Optimización Multiobjetivo en Diseño de Sistemas de Seguridad	157
6.3.1	Diseño Óptimo de Sistemas de Seguridad	157
6.3.2	Aplicación práctica: Sistema de Contención por Inyección en Spray	159
6.3.3	Datos adoptados	160

6.3.4	Configuración de parámetros	161
6.3.5	Métricas	161
6.3.6	Análisis de resultados con TOPSISGA	163
6.3.7	Análisis de resultados con WP-TOPSISGA	164
6.3.8	Análisis de resultados con CMOEA&TOPSIS	164
6.3.9	Análisis de resultados con CMOEA20-80&TOPSIS	168
6.3.10	Conclusiones	171
7	Conclusiones y Líneas Futuras	175
7.1	Introducción	175
7.2	Resumen de las Propuestas	177
7.3	Líneas Futuras de Investigación	179
	Bibliografía	182

Índice de Tablas

1.1	Los roles del <i>decisor</i> y del <i>hombre de estudio</i>	10
1.2	Matriz de decisión para un conjunto A de acciones y un conjunto G de criterios.	25
1.3	Matriz de decisión del problema de Vallin y Vanderpooten.	26
1.4	Datos normalizados, acciones ideal y anti-ideal en el problema de Vallin y Vanderpooten.	31
1.5	Distancias Euclídeas, similaridad y ordenaciones según TOPSIS y suma ponderada.	32
1.6	Comparaciones entre acciones para el criterio g_1 (izquierda) y comparaciones entre acciones para el criterio g_2 (derecha).	36
1.7	Resumen de los conjuntos J^+ (izquierda), $J^=$ (centro) y J^- (derecha).	36
1.8	Resumen de coeficientes W^+ (izquierda), $W^=$ (centro) y W^- (derecha).	36
1.9	Coeficientes de concordancia C_{jn} (zquierda) y discordancia D_{jn} (derecha).	37
2.1	Resultados obtenidos al aplicar el método de los pesos (y variar el vector pesos w_i) al problema F_2 de Schaffer.	49
2.2	Resultados obtenidos al aplicar el método de la distancia a un punto de referencia (y variar el vector pesos w_i) al problema F_2 de Schaffer.	52
2.3	Resultados obtenidos al aplicar el método de las ϵ -restricciones (y variar el vector ϵ) al problema F_2 de Schaffer.	55
3.1	Tabla comparativa entre paradigmas	67
4.1	Soluciones de ZDT5.	110
5.1	Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes alcanzados con TOPSISGA (frente B) y NSGAI (frente A).	116
5.2	Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes alcanzados con WP-TOPSISGA (frente B) y R-NSGAI (frente A).	120
5.3	Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes alcanzados con CMOEA&TOPSIS (frente B) y TOPSISGA (frente A).	129

5.4	Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes alcanzados con CMOEA&TOPSIS (frente B) y NSGAI (frente A).	130
5.5	Resultados obtenidos en la comparativa de los frentes alcanzados con CMOEA&TOPSIS y NSGAI, en base al hipervolumen H (métrica S).	130
5.6	Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes alcanzados con CMOEA20-80&TOPSIS (frente B) y NSGAI (frente A).	134
5.7	Resultados obtenidos en la comparativa de los frentes alcanzados con CMOEA20-80&TOPSIS y NSGAI, según el hipervolumen H (métrica S).	134
6.1	Pesos y beneficios adoptados para los objetos en la mochila 1 y en la mochila 2.	139
6.2	Valores en tanto por ciento de la métrica relativa $C(A,B)$, obtenidos al comparar los frentes alcanzados con TOPSISGA (frente B) y NSGAI (frente A).	142
6.3	Valores en tanto por ciento de la métrica relativa C , obtenidos cuando se comparan los frentes alcanzados (en la R_d -PF) con WP-TOPSISGA (frente B) y R-NSGAI (frente A).	145
6.4	Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes alcanzados con CMOEA&TOPSIS (frente B) y TOPSISGA (frente A).	148
6.5	Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes completos alcanzados con CMOEA&TOPSIS (frente B) y NSGAI (frente A).	151
6.6	Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes (RP) alcanzados con CMOEA&TOPSIS (frente B) y NSGAI (frente A).	151
6.7	Valores de la métrica relativa S (hipervolumen H), obtenidos al comparar los frentes completos alcanzados con CMOEA&TOPSIS y NSGAI.	153
6.8	Valores porcentuales de la métrica relativa C , obtenidos al comparar los frentes completos alcanzados con CMOEA20-80&TOPSIS (frente B) y NSGAI (frente A).	153
6.9	Valores porcentuales de la métrica relativa C , obtenidos al comparar los frentes (RP) alcanzados con CMOEA20-80&TOPSIS (frente B) y NSGAI (frente A).	154
6.10	Valores de la métrica relativa S (hipervolumen H), obtenidos al comparar los frentes completos alcanzados con CMOEA20-80&TOPSIS y NSGAI.	154
6.11	Valores de parámetros del problema SCIS.	160
6.12	Valores soluciones del problema de SCIS.	161

6.13	Valores en tanto por ciento de la métrica relativa $C(A,B)$, obtenidos al comparar los frentes alcanzados con TOPSISGA (frente B) y NSGAI (frente A) con el problema de SCIS.	163
6.14	Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes alcanzados con WP-TOPSISGA (frente B) y R-NSGAI (frente A).	164
6.15	Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes completos alcanzados con CMOEA&TOPSIS (frente B) y TOPSISGA (frente A) con el problema SCIS.	169
6.16	Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes completos alcanzados con CMOEA&TOPSIS (frente B) y NSGAI (frente A) con el problema SCIS.	169
6.17	Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes (RP) alcanzados con CMOEA&TOPSIS (frente B) y NSGAI (frente A) con el problema SCIS.	170
6.18	Valores de la métrica relativa S (hipervolumen H), obtenidos al comparar los frentes completos alcanzados con CMOEA&TOPSIS y NSGAI con el problema de SCIS.	170
6.19	Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes completos alcanzados con CMOEA20-80&TOPSIS (frente B) y NSGAI (frente A) con el problema de SCIS.	171
6.20	Problema de SCIS. Porcentajes de la métrica C , de comparar los frentes (RP) alcanzados con CMOEA20-80&TOPSIS (frente B) y NSGAI (frente A).	172
6.21	Valores de la métrica relativa S (hipervolumen H), obtenidos al comparar los frentes completos alcanzados con CMOEA20-80&TOPSIS y NSGAI con el problema de SCIS.	172

Índice de Figuras

1.1	La realidad humana (izquierda) y modelo asociado (derecha).	9
1.2	Representación matricial de la relación binaria R sobre el conjunto A (izquierda) y del grafo (A,R) (derecha).	14
1.3	Convenciones gráficas para la preferencia P, la indiferencia I y la incomparabilidad J.	16
1.4	Representación del Preorden total (I,P): matricial (izquierda) y gráfica (derecha).	18
1.5	Representación del Orden total (I,P): matricial (izquierda) y gráfica (derecha).	19
1.6	Preorden parcial (I,P,J): representación matricial (izquierda), gráfica (centro) y dominancia debil de Pareto (derecha).	20
1.7	Orden parcial (I,P,J): representación matricial (izquierda), gráfica (centro) y dominancia de Pareto (derecha).	21
1.8	Representación de las acciones del problema de Vallin y Vanderpooten (izquierda) y test de dominancia (derecha).	26
1.9	Evolución de la suma ponderada en función del valor w_1 , para las cuatro acciones del problema de Vallin y Vanderpooten.	27
1.10	Acción b fuera de la combinación convexa a-c en el problema de Vallin y Vanderpooten.	28
1.11	Situación general en el método TOPSIS, con a_j como mejor acción (izquierda) y dilema de trabajar con la solución a_j o a_z (derecha).	29
1.12	Acciones ideal (I^+) y anti-ideal (I^-).	30
1.13	Grafo de relaciones de superación entre las acciones b y d.	37
2.1	Clasificación de los métodos de Optimización Multiobjetivo.	40
2.2	Espacio de búsqueda (izquierda). Espacio realizable (centro). Espacio de los objetivos (derecha).	43
2.3	Relaciones posibles de dominancia entre soluciones.	43
2.4	Concepto de ϵ -dominancia (izquierda). Dominancia de Pareto (derecha).	45
2.5	Puntos Ideal y Nadir (izquierda). Punto de Funcionamiento (PF) y Región dominante del Punto de Funcionamiento (R_d -PF) (derecha).	46
2.6	Interpretación gráfica del método de los pesos (izquierda). Dificultad para el método de los pesos (derecha).	48
2.7	Problema F_2 de Schaffer.	49

2.8	Ilustración gráfica en el espacio de los objetivos, de las soluciones obtenidas al aplicar el método de los pesos al problema F_2 de Schaffer.	50
2.9	Interpretación gráfica del método de la distancia a un punto de referencia.	52
2.10	Ilustración gráfica en el espacio de los objetivos, de las soluciones obtenidas al aplicar el método de la distancia a un punto de referencia al problema F_2 de Schaffer.	53
2.11	Ilustración gráfica del funcionamiento del método de las ϵ -restricciones.	54
2.12	Ilustración gráfica en el espacio de los objetivos, de las soluciones obtenidas al aplicar el método de las ϵ -restricciones al problema F_2 de Schaffer.	55
2.13	Ilustración gráfica de inconvenientes del método de las ϵ -restricciones.	56
3.1	Ilustración de un cromosoma binario (izquierda) y un cromosoma real (derecha).	67
3.2	Ilustración de un cromosoma binario de tres genes que codifican las variables x_1 , x_2 y x_3 .	68
3.3	Ilustración de la decodificación del genotipo al fenotipo.	69
3.4	Selección RWS (izquierda): se lanza la ruleta N veces y cada vez se selecciona el individuo marcado. Selección SUS (derecha): se lanza la ruleta una sola vez y se seleccionan los N individuos marcados.	73
3.5	Representación esquemática del cruce de un punto.	74
3.6	Representación esquemática del cruce de dos puntos.	74
3.7	Representación esquemática del cruce uniforme.	75
3.8	Representación esquemática de una mutación en un cromosoma.	75
3.9	Generación de la población inicial: la figura izquierda en comparación con la figura derecha, muestra una situación más favorable para la búsqueda del óptimo.	77
3.10	Cálculo de la distancia $d_{1,2}$ de Hamming en el espacio del genotipo.	78
3.11	Cálculo de la distancia $d_{1,2}$ en el espacio de la decisión.	79
4.1	Ilustración de dos objetivos fundamentales de un AEMO: convergencia y diversidad.	82
4.2	Ilustración de un primer inconveniente en problemas reales discretos: el tamaño N de la población no-dominada adoptado es inferior al tamaño M (desconocido) del frente de Pareto.	83
4.3	Ilustración de un segundo inconveniente en problemas reales discretos: en una región poblada, una solución Pareto óptima es eliminada, conservándose otra que pese a ser no-dominada en la actual generación no es Pareto óptima.	84
4.4	Asignación de rangos según Goldberg (izquierda) y según Fonseca y Fleming (derecha).	86
4.5	Asignación de rangos en base a la ϵ -dominancia de Laumanns.	87

4.6	La distancia de agrupamiento (crowding distance) utilizado en NSGAI (izquierda) y el más próximo vecino (the nearest neighbour) usado en SPEAI (derecha).	88
4.7	Proceso de decisión y AEMOs según Salazar et al.	90
4.8	Diagrama de flujo de RWGA.	92
4.9	Esquema de funcionamiento de VEGA.	93
4.10	Asignación de la aptitud a un individuo en MOGA (se considera una función lineal).	94
4.11	Esquema de funcionamiento de NSGAI.	96
4.12	Esquema de funcionamiento de ϵ -MOEA.	97
4.13	Problema en la utilización de la métrica C: $C(A,B)=C(B,A)=\frac{3}{5}$	106
4.14	Ilustración del hipervolumen H del frente A, limitado por el punto de referencia R.	107
4.15	Ilustración del ϵ -Indicador binario.	107
4.16	Vector solución de ZDT5.	109
4.17	Solución gráfica de ZDT5.	110
5.1	Esquema de funcionamiento para TOPSISGA, WP-TOPSISGA, NSGAI, CMOEA&TOPSIS y CMOEA20-80&TOPSIS.	114
5.2	Frente óptimo y preorden (según TOPSIS) encontrados con TOPSISGA: $N=10$, $w_1=0.0001$ y $w_2=0.9999$ (izquierda) y $w_1=0.001$ y $w_2=0.999$ (derecha).	115
5.3	Frente óptimo y preorden (según TOPSIS) encontrados con TOPSISGA: $N=10$, $w_1=0.01$ y $w_2=0.99$ (izquierda) y $w_1=0.09$ y $w_2=0.91$ (derecha).	115
5.4	Frente óptimo y preorden (según TOPSIS) encontrados con TOPSISGA: $N=10$, $w_1=0.5$ y $w_2=0.5$ (izquierda) y $w_1=0.99$ y $w_2=0.01$ (derecha).	115
5.5	Comparación de los frentes alcanzados por TOPSISGA y NSGAI con $N=10$ individuos. Además, para TOPSISGA se muestra el preorden (según TOPSIS) encontrado en las soluciones cuando los valores de los pesos son $w_1=w_2=0.5$.	116
5.6	Visualización gráfica de los resultados de la tabla 5.1.	116
5.7	Estimación de los pesos w_i según WP-TOPSISGA.	119
5.8	Frente óptimo ($N=10$ individuos) según WP-TOPSISGA y R-NSGAI.	121
5.9	Frente óptimo ($N=15$ individuos) según WP-TOPSISGA y R-NSGAI.	121
5.10	Frente óptimo ($N=20$ individuos) según WP-TOPSISGA y R-NSGAI.	121
5.11	Frente óptimo ($N=10$ individuos) según WP-TOPSISGA y R-NSGAI.	122
5.12	Frente óptimo ($N=15$ individuos) según WP-TOPSISGA y R-NSGAI.	122
5.13	Frente óptimo ($N=20$ individuos) según WP-TOPSISGA y R-NSGAI.	122
5.14	Frente óptimo ($N=10$ individuos) según WP-TOPSISGA y R-NSGAI.	123
5.15	Frente óptimo ($N=15$ individuos) según WP-TOPSISGA y R-NSGAI.	123
5.16	Frente óptimo ($N=20$ individuos) según WP-TOPSISGA y R-NSGAI.	123

5.17	Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS: N=15, SP=10, $w_1=0.0001$ y $w_2=0.9999$ (izquierda) y $w_1=0.001$ y $w_2=0.999$ (derecha).	127
5.18	Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS: N=15, SP=10, $w_1=0.01$ y $w_2=0.99$ (izquierda) y $w_1=0.09$ y $w_2=0.91$ (derecha).	127
5.19	Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS: N=15, SP=10, $w_1=0.5$ y $w_2=0.5$ (izquierda) y $w_1=0.99$ y $w_2=0.01$ (derecha).	127
5.20	Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS: N=20, SP=10, $w_1=0.0001$ y $w_2=0.9999$ (izquierda) y $w_1=0.001$ y $w_2=0.999$ (derecha).	128
5.21	Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS: N=20, SP=10, $w_1=0.01$ y $w_2=0.99$ (izquierda) y $w_1=0.09$ y $w_2=0.91$ (derecha).	128
5.22	Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS: N=20, SP=10, $w_1=0.5$ y $w_2=0.5$ (izquierda) y $w_1=0.99$ y $w_2=0.01$ (derecha).	128
5.23	Visualización gráfica de la tabla 5.3.	129
5.24	Comparación de los frentes alcanzados por CMOEA&TOPSIS (se omite la clasificación TOPSIS inducida) (N=15, SP=10 y $w_1=w_2=0.5$) con TOPSISGA (N=10 y $w_1=w_2=0.5$) (izquierda) y con NSGAIL (N=15) (derecha).	129
5.25	Visualización gráfica de la tabla 5.4.	130
5.26	Visualización gráfica de la tabla 5.5.	130
5.27	Distribución de las rentas de una nación y valores críticos según la regla 20-80 de Pareto.	133
5.28	Visualización gráfica de las tablas 5.6 (izquierda) y 5.4 (derecha). . .	134
5.29	Visualización gráfica de las tablas 5.7 (izquierda) y 5.5 (derecha). . .	134
6.1	Frontera de Pareto del problema de la Mochila Multiobjetivo (2 mochilas).	139
6.2	Frente óptimo y preorden logrados por TOPSISGA: N=10, $w_1=0.9$, $w_2=0.1$ (izquierda) y $w_1=0.75$, $w_2=0.25$ (derecha) con el problema de la Mochila Multiobjetivo.	141
6.3	Frente óptimo y preorden logrados por TOPSISGA: N=10, $w_1=w_2=0.5$ (izquierda) y $w_1=0.1$, $w_2=0.9$ (derecha) con el problema de la Mochila Multiobjetivo.	141
6.4	Visualización de los frentes alcanzados (N=10 individuos) por TOPSISGA ($w_1=0.9$ y $w_2=0.1$) y NSGAIL.	141
6.5	Vista gráfica de los valores de la tabla 6.2.	142
6.6	Frentes alcanzados (N=15) con WP-TOPSISGA y R-NSGAIL.	143
6.7	Frentes alcanzados (N=15) con WP-TOPSISGA y R-NSGAIL.	143

6.8	Frentes alcanzados (N=15) con WP-TOPSISGA y R-NSGAIL.	143
6.9	Frentes óptimos alcanzados (N=15 individuos) con WP-TOPSISGA y R-NSGAIL.	144
6.10	Frentes óptimos alcanzados (N=15 individuos) con WP-TOPSISGA y R-NSGAIL.	144
6.11	Resultados gráficos la tabla 6.3 con PF=($f_1=4097, f_2=3395$).	146
6.12	Resultados gráficos la tabla 6.3 con PF=($f_1=4180, f_2=3395$).	146
6.13	Resultados gráficos la tabla 6.3 con PF=($f_1=3806, f_2=3758$).	146
6.14	Resultados gráficos la tabla 6.3 con PF=($f_1=3392, f_2=3836$).	148
6.15	Resultados gráficos la tabla 6.3 con PF=($f_1=3581, f_2=3930$).	148
6.16	Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS para SP=10, $w_1=0.9999$ $w_2=0.0001$ y N=15 (izquierda), N=20 (derecha). 148	
6.17	Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS para SP=10, $w_1=0.9$ $w_2=0.1$ y N=15 (izquierda), N=20 (derecha). . .	149
6.18	Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS para SP=10, $w_1=0.75$ $w_2=0.25$ y N=15 (izquierda), N=20 (derecha). 149	
6.19	Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS para SP=10, $w_1=w_2=0.5$ y N=15 (izquierda), N=20 (derecha).	149
6.20	Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS para SP=10, $w_1=0.25$ $w_2=0.75$ y N=15 (izquierda), N=20 (derecha). 150	
6.21	Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS para SP=10, $w_1=0.1$ $w_2=0.9$ y N=15 (izquierda), N=20 (derecha). . .	150
6.22	Visualización gráfica de la tabla 6.4.	150
6.23	Visualización gráfica de la tabla 6.5.	151
6.24	Visualización gráfica de la tabla 6.6.	151
6.25	Visualización gráfica de la tabla 6.7.	153
6.26	Visualización gráfica de las tablas 6.8 (izquierda) y 6.5 (derecha). . .	153
6.27	Visualización gráfica de las tablas 6.9 (izquierda) y 6.6 (derecha). . .	154
6.28	Visualización gráfica de las tablas 6.10 (izquierda) y 6.7 (derecha). . .	154
6.29	Gráfica para interpretación de los resultados obtenidos en la comparación de CMOEA&TOPSIS con NSGAIL: N=15, $w_1=0.9$ $w_2=0.1$ (izquierda) y $w_1=0.75$ $w_2=0.25$ (derecha).	156
6.30	Gráfica para interpretación de los resultados obtenidos en la comparación de CMOEA&TOPSIS con NSGAIL: N=15, $w_1=0.5$ $w_2=0.5$ (izquierda) y $w_1=0.1$ $w_2=0.9$ (derecha).	156
6.31	Arbol simplificado de fallos del SCIS de una central nuclear.	158
6.32	Diagrama simplificado del SCIS de una central nuclear.	158
6.33	Diagrama del SCIS de una central nuclear utilizado como aplicación práctica.	159
6.34	Frontera de Pareto del problema de SCIS.	160
6.35	Frente óptimo y preorden logrados por TOPSISGA: N=10, $w_1=0.001$, $w_2=0.999$ (izquierda) y $w_1=0.005$, $w_2=0.995$ (derecha) con el problema de SCIS.	162

6.36	Frente óptimo y preorden logrados por TOPSISGA: $N=10$, $w_1=w_2=0.5$ (izquierda) y $w_1=0.99$, $w_2=0.01$ (derecha) con el problema de SCIS. . .	162
6.37	Frente alcanzado por NSGAI con el problema de SCIS.	162
6.38	Vista gráfica de los valores de la tabla 6.13	163
6.39	Frente alcanzado ($N=10$) con WP-TOPSISGA con el problema de SCIS.	165
6.40	Frente alcanzado ($N=10$) con WP-TOPSISGA con el problema de SCIS.	165
6.41	Problema de SCIS. Frente óptimo y ordenación TOPSIS logrados con CMOEA&TOPSIS: $SP=10$, $w_1=0.001$ $w_2=0.999$ y $N=15$ (izquierda), $N=20$ (derecha).	166
6.42	Problema de SCIS. Frente óptimo y ordenación TOPSIS logrados con CMOEA&TOPSIS: $SP=10$, $w_1=0.005$ $w_2=0.995$ y $N=15$ (izquierda), $N=20$ (derecha).	166
6.43	Problema de SCIS. Frente óptimo y ordenación TOPSIS logrados con CMOEA&TOPSIS: $SP=10$, $w_1=0.01$ $w_2=0.99$ y $N=15$ (izquierda), $N=20$ (derecha).	166
6.44	Problema de SCIS. Frente óptimo y ordenación TOPSIS logrados con CMOEA&TOPSIS: $SP=10$, $w_1=w_2=0.5$ y $N=15$ (izquierda), $N=20$ (derecha).	167
6.45	Problema de SCIS. Frente óptimo y ordenación TOPSIS logrados con CMOEA&TOPSIS: $SP=10$, $w_1=0.9$ $w_2=0.1$ y $N=15$ (izquierda), $N=20$ (derecha).	167
6.46	Problema de SCIS. Frente óptimo y ordenación TOPSIS logrados con CMOEA&TOPSIS: $SP=10$, $w_1=0.99$ $w_2=0.01$ y $N=15$ (izquierda), $N=20$ (derecha).	167
6.47	Visualización gráfica de la tabla 6.15.	169
6.48	Visualización gráfica de la tabla 6.16.	169
6.49	Visualización gráfica de la tabla 6.17.	170
6.50	Visualización gráfica de la tabla 6.18.	170
6.51	Visualización gráfica de la tabla 6.19.	171
6.52	Visualización gráfica de la tabla 6.20.	172
6.53	Visualización gráfica de la tabla 6.21.	172

Lista de Acrónimos

AE	Algoritmo Evolutivo
AEMO	Algoritmo Evolutivo Multiobjetivo
AG	Algoritmo Genético
AHP	Analytic Hierarchy Process (Proceso de las Jerarquías Analíticas)
AMD	Ayuda Multicriterio a la Decisión
EEs	Estrategias Evolutivas
ELECTRE	Elimination Et Choix Traduisant la Realité (Eliminación y Elección para Representar la Realidad)
ϵ-MOEA	epsilon-Multiobjective Evolutionary Algorithm (Algoritmo Evolutivo Multiobjetivo-epsilon)
ϵ-NSGAI	epsilon-Non-Dominated Sorting Genetic Algorithm (Algoritmo Genético de Ordenación epsilon-No-Dominada)
HLGA	Hajela & Lin Genetic Algorithm (Algoritmo Genético de Hajela y Lin)
MAUT	Multi-Attribute Utility Theory (Teoría de la Utilidad Multi-Atributo)
MCDM	Multiple Criteria Decision Making (Toma de Decisiones Multicriterio)
MOGA	Multi-Objective Genetic Algorithm (Algoritmo Genético Multi-Objetivo)
NSGA	Non-Dominated Sorting Genetic Algorithm (Algoritmo Genético de Ordenación No-Dominada)
NSGA II	Elitist Non-Dominated Sorting Algorithm (Algoritmo Genético de Ordenación No-Dominada Elitista)
ODSS	Optimización de Diseño de Sistemas de Seguridad
OEM	Optimización Evolutiva Multiobjetivo
ORESTE	Organisation Rangement Et Synthèse des données relationnelles (Organización, Ordenación y Síntesis de Datos Relacionales)
PAES	Pareto Archived Evolution Strategy (Estrategia Evolutiva de Archivo de Pareto)

PE	Programación Evolutiva
PG	Programación Genética
PM	Problema Multiobjetivo
POM	Problema de Optimización Multiobjetivo
PROMETHEE	Preference Ranking Organisation Method for Enrichment Evaluations (Método de Organización y Clasificación de Preferencias para Enriquecimiento de las Evaluaciones)
PF	Punto de Funcionamiento
RP	Región de Precisión
R_d-PF	Región dominante del Punto de Funcionamiento
RD-NSGAI	Reference Direction Based NSGAI (Algoritmo Genético de Ordenación No-Dominada Elitista Basado en una Dirección de Referencia)
R-NSGAI	Reference Point Based NSGAI (Algoritmo Genético de Ordenación No-Dominada Elitista Basado en un Punto de Referencia)
RWGA	Random Weighted Genetic Algorithm (Algoritmo Genético de Ponderación Aleatoria)
SCIS	Sistema de Contención por Inyección en Spray
SMART	Similarity Measured Anchored Ranking Technique (Técnica de Clasificación Basada en la Medida de la Similitud)
SP	Soluciones de Precisión
SPEA	Strength Pareto Evolutionary Algorithm (Algoritmo Evolutivo de Fuerza de Pareto)
SPEAII	Strength Pareto Evolutionary Algorithm v.2 (Algoritmo Evolutivo de Fuerza de Pareto versión 2)
TACTIC	Traitement des Alternatives Compte Tenu de l'Importance des Critères (Tratamiento de Alternativas Considerando la Importancia de los Criterios)
TOPSIS	Technique for Order Preference by Similarity to an Ideal Solution (Técnica de Ordenación de Preferencias por Similitud a la Solución Ideal)
UTA	Utilités Additives (Utilidades Aditivas)
VEGA	Vector Evaluated Genetic Algorithm (Algoritmo Genético de Evaluación Vectorial)

Agradecimientos

Quisiera dar las gracias,

a Blas Galván que me dio la oportunidad de la realización de esta Tesis Doctoral y me introdujo en el mundo de los algoritmos genéticos. Recordaré siempre la primera reunión en su despacho del CEANI y tras darme una primera charla, sorprendido y algo incrédulo le pregunté: ¿pero estás seguro de que ésto optimiza?, “pues sí”, me respondió. Fue siempre muy generoso con su tiempo, apoyo y sugerencias. Me proporcionó todos sus códigos fuentes sobre algoritmos genéticos y me dejó toda libertad para hacer mis propias elecciones en todas las investigaciones. Deseo que encuentre en este trabajo, el porqué de justificar su confianza.

A David Greiner por su dedicación, su continuo apoyo y ayuda inestimable. Siempre encontró el tiempo necesario para responder a todas mis dudas y preguntas. Su espíritu crítico y constructivo, han influenciado mucho las orientaciones seguidas en esta tesis.

A Daniel Salazar por su ayuda en las distintas investigaciones que he llevado a cabo en estos años y horas de discusión sobre multiobjetivo y toma de decisiones.

A Javier Sánchez por su inestimable ayuda en todo lo relacionado con el L^AT_EX.

A todos los miembros de la división de Computación Evolutiva (CEANI), del Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (IUSIANI) de la Univeridad de Las Palmas de Gran Canaria, que me ayudaron en cualquier aspecto de la programación y el lenguaje C.

A Donghai por su paciencia.

Y de manera muy especial a toda mi familia.

Introducción general

Presentación

El *Paradigma Decisional Clásico* o *Unicriterio*, ha sido el modelo tradicionalmente utilizado en los procesos de toma de decisiones. Según este Paradigma, cuando un decisor debe decidir entre varias acciones cual es la mejor, se procede de la siguiente forma: primero se define el conjunto de acciones realizables para el problema en cuestión, a continuación y en base a un único criterio, que define las preferencias del decisor, se asocia a cada acción un número que representa el grado de preferencia de la misma, y finalmente utilizando técnicas más o menos sofisticadas, se procede a buscar entre las acciones realizables, aquella que tiene asociado el mejor valor o grado de preferencia. Este modelo posee gran solidez, la solución del problema es óptima y queda totalmente determinada por su formulación, no por el modo de resolución del mismo. El decisor, si estima que el modelo adoptado es coherente con la realidad, debe adoptarla.

Sin embargo, un modelo *Unicriterio* se desvía considerablemente de los procesos del mundo real, pues la mayoría de los problemas de decisión, sean económicos, industriales, financieros, políticos u otros, son *Multicriterio*, esto es, los agentes decisores implicados no están interesados en tomar sus decisiones con respecto a un único criterio, y sí más bien con arreglo a diferentes criterios usualmente en conflicto, que reflejen sus preferencias. Por ejemplo, el consejo de administración de una empresa puede no estar interesado en encontrar aquella acción que únicamente maximiza el beneficio, pero si aquella que armoniza criterios como beneficio, riesgo, empleo, etc.

Este nuevo punto de vista de ver y modelar la realidad es conocido como *Paradigma Multicriterio*, y por el se interesa la *Ayuda Multicriterio a la Decisión*. Inicialmente fue utilizada en las escuelas de economía y dirección de empresas. Aquí, el problema tipo consiste en *ayudar a elegir a un decisor, según varios criterios, entre varias acciones*.

Gradualmente, la *Ayuda Multicriterio a la Decisión*, se ha ido extendiendo a las ciencias y las ingenierías, donde más que la *decisión*, lo que se investiga es la *optimización*. Aquí, lo que se pretende es *buscar con rápidos algoritmos, un conjunto*

de soluciones eficientes. Ahora se prefiere hablar de *Optimización Multiobjetivo*. Si las preferencias del decisor son incluidas antes de comenzar el proceso búsqueda, se habla de *Optimización Multiobjetivo a Priori*, si por el contrario el decisor elige una solución, a partir del conjunto de soluciones eficientes final encontrado, entonces se habla de *Optimización Multiobjetivo a Posteriori*.

Cuando nos enfrentamos a problemas de *Optimización Multiobjetivo*, nos encontramos que la mayoría de ellos son calificados como *difíciles*. Un problema de optimización con variables continuas, se dice *difícil*, cuando no se conoce algoritmo capaz de identificar un óptimo global en un número finito de cálculos. Si se trata de un problema de optimización con variables discretas, se habla de problema *NP-difícil*, cuando la resolución exacta no es posible en un tiempo proporcional a N^n , siendo N el número de parámetros desconocidos del problema, y n una constante entera. Numerosos métodos han sido desarrollados con la intención de dar respuesta a este tipo de problemas. Métodos exactos tales como el Branch and Bound [180] [204], el Algoritmo A* [186] y la Programación Dinámica [24], han sido utilizados para la resolución de problemas con dos objetivos y reducidos tamaños. Sin embargo, con problemas de tamaños grandes y/o problemas en los que la toma de decisiones se quiere efectuar en base a más de dos objetivos, se recurre a métodos heurísticos. Estos métodos no garantizan encontrar de forma exacta, el conjunto de soluciones óptimas (o frente de Pareto), pero sí, con una rapidez importante, un conjunto aproximado (o frente de Pareto aproximado). Diferentes heurísticas han sido propuestas para la resolución de problemas de *Optimización Multiobjetivo*: *Algoritmos Evolutivos* [66][85][98][154], *Búsqueda Tabú* [83][84], *Recocido Simulado* [111], entre otros.

Los *Algoritmos Evolutivos*, inspirados en la evolución natural, han revelado su efectividad en la resolución de problemas de *Optimización Multiobjetivo* del mundo real. Al trabajar sobre poblaciones con múltiples potenciales soluciones, los *Algoritmos Evolutivos* se revelan muy aptos en la exploración de espacios extensos y poco tratables para los algoritmos tradicionales. Esta particularidad hace posible que se adapten muy bien a la búsqueda de soluciones del *frente de Pareto*, mostrándose capaces de hallar un conjunto de soluciones de Pareto con sólo unas pocas ejecuciones del algoritmo. Además, los *Algoritmos Evolutivos* han demostrado una gran capacidad, cuando trabajan con problemas ya sean discretos, continuos, mixtos, con frentes de soluciones convexos, no convexos, etc.

Todo esto, constituye el motivo para que en esta tesis, nos hayamos interesado por los *Algoritmos Evolutivos Multiobjetivo a Priori*. En este marco de trabajo, se requiere que el decisor exprese sus preferencias, para que a continuación sean incluidas en el proceso de búsqueda del conjunto de *Pareto*. Para la ilustración del funcionamiento, y evaluación de las eficiencias de las contribuciones algorítmicas *Evolutivas Multiobjetivo* que se hacen, se escogieron problemas test y problemas con aplicaciones prácticas

a las ciencias e ingeniería, en el dominio del *discreto*. En concreto, como problema test se eligió el *ZDT5* de Zitzler, Deb y Thiele [220]. Como problemas de aplicaciones prácticas se eligieron dos. El primero, es el problema de la *Mochila Multiobjetivo* en variables binarias (en inglés 0-1MOKP) [105][107][128][135][174][219]. Este problema modela aplicaciones reales en campos como la ingeniería, la economía, las finanzas, etc. El segundo de los problemas, consiste en la *Optimización Multiobjetivo en Diseño de Sistemas de Seguridad* (ODSS), en particular se utilizó el *Sistema de Contención por Inyección en Spray* (SCIS) [78][89][91], de una central nuclear de generación eléctrica. Se trata de un problema muy estudiado en la división de Computación Evolutiva y Aplicaciones (CEANI) del Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (IUSIANI) de la Universidad de Las Palmas de Gran Canaria (ULPGC).

Resultados y contribuciones

En el capítulo 5 de contribuciones se detallan los algoritmos propuestos, explicando cuáles son las contribuciones que se hacen y qué pruebas numéricas se llevaron a término para la evaluación de la eficiencia. Las contribuciones más importantes que destacamos son los siguientes:

- TOPSISGA, trata de dar respuesta a algunos de los inconvenientes que pueden producirse con un AEMO al tratar con problemas reales multiobjetivo discretos. Por ejemplo, si el número de soluciones eficientes del problema es muy numeroso y el tamaño de la población no-dominada está limitado, el AEMO no puede contener el conjunto completo de soluciones, obligando a que se pierdan soluciones potenciales de interés para el decisor. La propuesta consiste en un AEMO que mezcla las metodologías evolutivas multiobjetivo (se utiliza NSGAI [51][55], aunque cualquier otra metodología evolutiva multiobjetivo puede ser usada) con el método bien conocido de Toma de Decisiones Multicriterio TOPSIS [101]. Dos ideas fundamentales subyacen en el método: la primera es guiar la búsqueda, de forma que se permita concentrar más soluciones válidas en aquella zona del frente óptimo que resulta del interés de un decisor; la segunda consiste en inducir en las soluciones encontradas un preorden completo (Problemática P_γ de Roy [161][163]), en base a la similaridad [45] al punto ideal.
- La propuesta WP-TOPSISGA, trata con problemas industriales, económicos u otros, caracterizados en el espacio de los objetivos por un Punto de Funcionamiento (PF) [134][136], el cual identifica una solución estable y realizable de trabajo no necesariamente eficaz o Pareto óptima. En este tipo de problemas, un decisor inmerso en la tarea de mejorar el PF, puede estar interesado

por un pequeño número de soluciones, por ejemplo, aquellas que localizadas en una cierta región del frente óptimo (la R_d -PF) [134] dominan el PF. En este propósito, WP-TOPSISGA combina la segunda generación de AEMOs (usamos NSGAI, aunque otro método evolutivo multiobjetivo de segunda generación puede ser utilizado), el método de Toma de Decisiones Multicriterio TOPSIS y el PF. El método es capaz de conducir la búsqueda de soluciones hacia la región de interés (la R_d -PF) y clasificar las soluciones halladas en base a la similaridad [45] al punto ideal. Es importante notar, que en el cálculo del valor de la similaridad, es necesario conocer los valores de los pesos w_i otorgados por un decisor acorde a sus preferencias, a los diferentes objetivos del problema. Sin embargo, con WP-TOPSISGA los valores de los pesos son estimados automáticamente en cada iteración del algoritmo en base al valor de las coordenadas del PF.

- Con CMOEA&TOPSIS, se persiguen fundamentalmente dos objetivos: (1) encontrar un mejor equilibrio exploración-explotación que el conseguido con TOPSISGA (este método busca una acentuada densidad de soluciones en una zona específica del frente óptimo, pero en contrapartida, aplica una fuerte presión de selección enfatizando mucho más aquellas soluciones de la zona de interés que al resto) y (2) dar solución a aquella situación en la que un decisor indica que parte de frente óptimo es de su interés, pero no está seguro de su elección. El método propuesto, une en un único algoritmo (algoritmo combo) las habilidades de los mejores AEMOs a posteriori de segunda generación (hemos utilizado NSGAI, si bien otro método puede ser adoptado) y las de los AEMOs a priori (hemos utilizado en este caso TOPSISGA, aunque otro método de toma de decisiones se puede utilizar). En el frente final de soluciones alcanzado por el algoritmo, se observan dos regiones bien diferenciadas, la primera, se caracteriza por una acentuada y constante densidad de soluciones (clasificadas según TOPSIS), de tamaño el número de Soluciones de Precisión (SP) que el decisor haya elegido, la segunda se distingue por su amplitud y diversidad de soluciones.
- En la propuesta CMOEA&TOPSIS anterior, se observa que si el valor SP elegido por el decisor es igual al tamaño N de la población de soluciones no-dominadas, entonces el comportamiento de CMOEA&TOPSIS equivale al de TOPSISGA, de otra parte, si el valor SP adoptado es cero, entonces el comportamiento de CMOEA&TOPSIS es análogo al de NSGAI. Con CMOEA20-80&TOPSIS se propone una solución a la cuestión: ¿Qué tamaño SP sería el idóneo elegir?. La ley 20-80 de Pareto [149] es para ello utilizada en cada iteración del algoritmo. El resultado final obtenido por CMOEA20-80&TOPSIS, consiste en un amplio y bien distribuido frente de soluciones pero que distingue una región, la de interés para el decisor, caracterizada por su alta densidad de soluciones y de tamaño

(SP), el estimado por el algoritmo según la ley 20-80 de Pareto.

Publicaciones realizadas

Durante el desarrollo de esta tesis se han realizado una serie de publicaciones que citamos a continuación:

- Optimización evolutiva multiobjetivo del diseño de Sistemas de Seguridad: Comparando el NSGAI y el criterio del tomador de decisiones [137].
- New Trends in Evolutionary Optimization and its Impact on Dependability Problems [211].
- Multiple-Objective Genetic Algorithm Using the Multiple Criteria Decision Making Method TOPSIS [135].
- Multiple-Objective Evolutionary Algorithms Using the Working Point and the TOPSIS Method [134].
- En las Cercanías del Punto de Funcionamiento Utilizando Algoritmos Genéticos Multiobjetivo [136].
- New Evolutionary Methodologies for Integrated Safety System Design and Maintenance [80].
- Optimización Conjunta del Diseño y la Estrategia de Mantenimiento de Sistemas de Seguridad en Industrias Potencialmente Peligrosas [79].

Organización del documento

Esta tesis está organizada en 7 capítulos. Los tres primeros capítulos introducen nociones y conceptos básicos, que se consideran claves para entender el trabajo de tesis. Los últimos cuatro capítulos refieren el trabajo de tesis y sus resultados.

El capítulo 1 presenta una introducción a la Ayuda Multicriterio a la Decisión. En primer lugar se discuten los Paradigmas Unicriterio y Multicriterio. Seguidamente se aborda la elaboración de un modelo de decisión (o proceso de ayuda a la decisión) de una realidad, describiéndose el conjunto de artefactos (las acciones, las problemáticas, el modelado de las preferencias, el modelado de las consecuencias, los criterios y la agregación de las evaluaciones) que produce el proceso. Se finaliza el capítulo, exponiendo con ejemplos sencillos, algunos de los métodos más conocidos de agregación completa y agregación parcial.

El capítulo 2 describe nociones básicas del dominio de la Optimización Multiobjetivo. Se comienza con una breve discusión de los conceptos Multiobjetivo y Multicriterio. A continuación, se presentan diferentes propuestas de clasificación de la Optimización Multiobjetivo hechas por autores relevantes en el dominio. Se sigue con la representación formal de un problema de Optimización Multiobjetivo y la descripción de algunos conceptos fundamentales como dominancia y puntos de referencia. Por último se exponen algunos métodos conocidos, ilustrados con ejemplos simples.

El capítulo 3 hace una introducción a los Algoritmos Evolutivos. Se comienza con una síntesis de los principios de la teoría evolutiva que los sustentan. Seguidamente se hace un breve recorrido histórico, presentando las primeras y posteriores versiones de Algoritmos Evolutivos introducidas por diferentes autores, así como el vocabulario particular utilizado en el dominio y paradigmas principales. Finalmente se describen los diferentes tipos de codificaciones de variables existentes, y algunos de los operadores genéticos más conocidos y utilizados.

El capítulo 4 presenta un exhaustivo estado del arte de los Algoritmos Evolutivos Multiobjetivo. Se hace un recorrido de los métodos más significativos desarrollados hasta el presente. Los métodos se han clasificado según tres tipos: métodos de transformación en un único objetivo, métodos No-Pareto y métodos Pareto. Una sección del capítulo, es enteramente consagrada a Algoritmos Evolutivos Multiobjetivo que introducen preferencias del decisor antes del inicio del proceso de búsqueda de soluciones. Se hace igualmente, una revisión de las métricas existentes en la evaluación y comparación de los algoritmos. Se finaliza presentando el problema test ZDT5 [220], el cual describe un problema engañoso para problemas de variables discretas binarias.

El capítulo 5 contiene las contribuciones originales aportadas en esta tesis, en el ámbito de los Algoritmos Genéticos Multiobjetivo a priori aplicados a problemas discretos. Se hace una explicación en profundidad de los algoritmos propuestos, además de una comparativa con otros Algoritmos Evolutivos Multiobjetivo. El problema test ZDT5 es usado, para comparar y probar los algoritmos.

El capítulo 6 aborda la aplicación de los métodos propuestos en el capítulo 5 de contribuciones a dos problemas del mundo real: el problema de la Mochila Multiobjetivo en variables binarias [105][107][128][135][174][219] y el problema de Optimización Multiobjetivo en Diseño de Sistemas de Seguridad (ODSS) (en particular se utilizó el Sistema de Contención por Inyección en Spray (SCIS) [78][89][91], de una central nuclear de generación eléctrica).

El capítulo 7 presenta las conclusiones, y se comentan, considerando los resultados obtenidos en los capítulos 5 y 6 de contribuciones y aplicaciones, algunas líneas futuras de investigación que podrían resultar de interés.

Capítulo 1

Ayuda Multicriterio a la Decisión

1.1 Paradigmas Unicriterio y Multicriterio

Los modelos clásicos de investigación de operaciones, establecidos progresivamente desde los años 1937, han analizado los procesos de toma de decisiones siguiendo el *Paradigma Decisional Clásico o Unicriterio*. Según este Paradigma, para decidir entre varias acciones¹, cual es la que más conviene, se procede del siguiente modo: primero se establece el conjunto de acciones realizables para el problema en cuestión, después basándose en un único criterio (que define las preferencias del decisor), se asocia a cada acción un número que representa el grado de preferencia de la misma y finalmente, utilizando determinadas técnicas matemáticas, se busca entre las acciones realizables, aquella que tiene asociado un mayor grado de preferencia. Un modelo *Unicriterio*, corresponde a un problema matemáticamente bien planteado, la solución del problema es óptima, quedando totalmente determinada por su formulación y no por el modo de resolución del mismo. La solución óptima de un problema que sigue un modelo *Unicriterio*, será siempre propuesta al decisor, y éste, si estima que el modelo es coherente con la realidad, no tiene más espacio de libertad, debe aceptarla.

De otra parte, en el mundo real, la mayoría de los problemas de decisión, sean económicos, industriales, financieros, políticos u otros, son *Multicriterio*, esto es, los agentes decisores implicados no toman sus decisiones basándose en un único criterio, sino que el proceso decisional está influenciado por una gama de criterios usualmente en conflicto. Por ejemplo, nadie compra un automóvil en base únicamente al precio (criterio financiero), otros criterios como el confort, la calidad, el resultado, el prestigio,... son sin duda siempre tenidos en cuenta. Además, no todo el mundo se manifiesta de la misma manera, de hecho, para alguien que mira diferentes automóviles, su elección está sujeta a sus gustos personales. Diferentes personas asignan desiguales conjuntos de pesos o preferencias a los criterios. Este enfoque, conocido

¹En AMD se utilizan indiferentemente los términos acción, alternativa o solución.

como *Paradigma Multicriterio*, resulta particularmente de interés en el tratamiento de realidades humanas, y por el se interesa la *Ayuda Multicriterio a la Decisión* (AMD). Su existencia es bastante reciente, suele situarse en 1972 con motivo de la celebración de la *First International Conference on MCDM* (Multiple Criteria Decisión Making) organizada por Cochrane y Zeleny [27] en la University of South Carolina, si bien los conceptos de *Análisis Multicriterio* y *Ayuda a la Decisión*, tienen orígenes más lejanos. En efecto, ya en el siglo XVIII, Borda [13] y Condorcet [46] se interesaron por los procedimientos de decisión colectiva. Los problemas económicos en los que varios agentes decisores realizan diferentes elecciones a menudo en conflicto, ocuparon a Pareto [149] a finales del siglo XIX. Entre finales de los años 1940 y principios de los años 1950, se presentaron varios trabajos fundamentales, entre los que se destaca la teoría de juegos de Nash [145][146], la teoría de la racionalidad limitada de Simon [182][183], el teorema sobre procedimientos dictatoriales de Arrow [4] y el problema de criterios múltiples en programación lineal abordado por Kuhn y Tucker [121]. Por último, entre finales de los años 1950 y hasta principios de los años 1970, se propusieron numerosos métodos que hoy en día son clásicos, destacamos por ejemplo la programación por metas propuesta por Charnes-Cooper [25][26], los métodos basados en relaciones de superación presentados por Benayoun [12] y Roy [159] y el método fundado en la teoría de la utilidad en presencia de criterios múltiples de Keeney y Raiffa [110]. Una amplia y detallada reseña histórica sobre los orígenes de la *Ayuda Multicriterio a la Decisión* se puede encontrar en [9][153][196].

Considerar varios criterios en el proceso de decisión, contribuye conforme a los deseos del decisor, a un planteamiento correcto del problema, pero desafortunadamente también corresponde a un problema cuyos resultados dependen no sólo de cómo ha sido formulado el problema, sino también del procedimiento utilizado para encontrarlos, además en general no ha de existir una única solución que sea la mejor en todos los criterios.

1.2 Problema de decisión

Un problema de decisión, se puede entender como cualquier problema del mundo real, planteado en un contexto particular, más o menos limitado que llamamos *realidad* (figura 1.1 izquierda). La realidad puede adoptar aspectos muy variados, por ejemplo puede ser de naturaleza económica, industrial, financiera, administrativa, medio ambiente, etc. Comprender la realidad no es un problema fácil, por ejemplo: los límites pueden ser difusos y estar mal definidos, es necesario conocer sus mecanismos de funcionamiento, una gran cantidad de datos están normalmente en juego, el mundo exterior hace presión sobre ella e inversamente ella puede crear factores de influencia en su entorno, puede ser definida de forma rígida o evolutiva en el tiempo, etc.

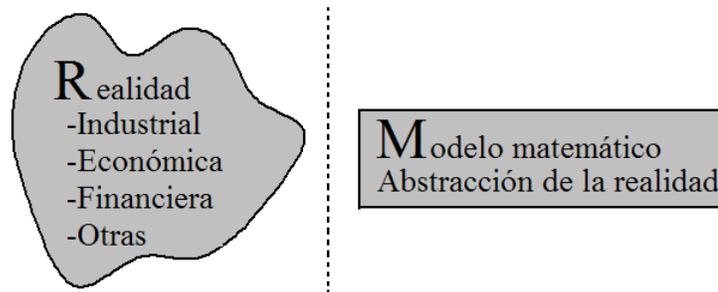


Figura 1.1: La realidad humana (izquierda) y modelo asociado (derecha).

Cuando se trata la realidad, esta puede ser abordada de forma cualitativa o cuantitativa. No obstante ambos enfoques no son incompatibles, de hecho y sobre todo si se trata de realidades muy complejas, el enfoque cuantitativo se desarrolla de más en más, complementándolo con el enfoque cualitativo.

El enfoque cualitativo consiste en observar la realidad y hacer uso del conocimiento y experiencia adquiridos, para finalmente decidir. La intuición aquí juega un papel muy importante. Hoy en día, las realidades económicas, industriales u otras, son frecuentemente de tal magnitud y complejidad, que resulta prácticamente imposible abordarlas de forma exclusivamente cualitativa.

El enfoque cuantitativo asocia a cada realidad, un modelo² matemático de decisión (figura 1.1 derecha), a través del cual resulta posible describir, comprender y manejar la realidad. Este enfoque resulta más delicado y más difícil, pues conviene desarrollar un lenguaje matemático particular, cuantificar los factores de influencia, analizar los mecanismos de funcionamiento y prever las consecuencias de las posibles decisiones. Todo modelo no será sin embargo más que una construcción humana, una abstracción de la realidad.

1.3 Actores que intervienen en un proceso de decisión

Para Roy [163], cuando se aborda la elaboración de un modelo de decisión (o proceso de decisión [196][155]) de una realidad, los actores que participan pueden ser varios:

- *Los que intervienen*: aquellos que en función de sus sistemas de valores hacen valer sus preferencias. Estos actores pueden ser un grupo responsable de una

²Para un observador, un modelo M es un modelo de una realidad R, si puede obtener estudiando M, las respuestas a las preguntas que se hace sobre R

Tabla 1.1: Los roles del *decisor* y del *hombre de estudio*

<i>Decisor</i>	<i>Hombre de estudio</i>
decisión	análisis
realidad	modelo
responsabilidad	ayuda, asistencia
factores de influencia	funciones, restricciones
objetivos	criterios de evaluación

empresa, un consejo de administración, un individuo,... que llamamos *decisor*.

- *Los afectados*: quienes se benefician o sufren las consecuencias de la decisión tomada. Como es importante que esta sea consensuada, conviene tener muy en cuenta sus preferencias.
- *El hombre de estudio*: quien ejerce la actividad de *Ayuda a la Decisión*³. El hombre de estudio puede componerse de un equipo que incluye ingenieros, economistas, sociólogos, etc.
- *El solicitante*: quien encarga el trabajo y se ocupa de proporcionar el apoyo y los medios que pueda necesitar el hombre de estudio.

Los actores que pueden participar en el proceso, interactúan continuamente con sus opiniones y criterios, pudiendo además surgir conflictos entre ellos.

Brans [19], precisa los roles y complementariedad entre el decisor y el hombre de estudio (tabla 1.1). La tabla muestra como el decisor, que trabaja en el mundo real, toma la decisión y la responsabilidad de dicha decisión, mientras el hombre de estudio, trabaja en el modelo, se encarga del análisis y procura ayuda y asistencia. Además, es el decisor quien tiene en cuenta los factores de influencia mientras el hombre de estudio los representará en el lenguaje matemático del modelo, bajo forma de funciones y restricciones. Por último, observamos que los objetivos que tiene el decisor, serán traducidos en criterios de evaluación por el hombre de estudio.

³Según Roy [163], la Ayuda a la Decisión es la actividad de aquel que, tomando apoyo sobre modelos claramente explicitados pero no necesariamente completamente formalizados, ayuda a obtener elementos de respuestas a las cuestiones que se plantean los que intervienen en un proceso de decisión, elementos que concurren a aclarar la decisión y a recomendar, un comportamiento que aporte coherencia entre la evolución del proceso y los objetivos de los que intervienen.

1.4 Elaboración del modelo de decisión

Según Roy [163], cuando el hombre de estudio se enfrenta a la construcción de un modelo de decisión, debe plantearse las siguientes cuestiones:

- ¿Cómo delimitar e identificar las acciones disponibles?.
- ¿Hacia que problemática se va a orientar el problema?.
- ¿Cómo se van a modelar las preferencias?.
- ¿Cuáles son las consecuencias de las posibles acciones?.
- ¿Cuáles son los criterios?.
- ¿Cómo agregar las evaluaciones?.

Estas preguntas, producen en el proceso un conjunto de artefactos: *las acciones, las problemáticas, el modelado de preferencias, el modelado de las consecuencias, los criterios y la agregación de las evaluaciones*. En las siguientes secciones se hace una discusión detallada de estos artefactos.

1.5 Las acciones

Una parte importante del proceso de toma de decisiones consiste, en conocer cuáles son las acciones involucradas. Para el ejemplo de compra de un automóvil dado en el apartado 1.1, quien decide la compra, precisa tener claro entre qué opciones (acciones) puede escoger. Las acciones no están dadas, hay que buscarlas lo cual no es precisamente una tarea sencilla, y por eso, no es extraño que un mismo problema pueda ser modelado con conjuntos de diferentes acciones. Según Roy [163], cuando se habla de acciones conviene realizar la siguiente distinción:

- *Acciones reales*: cuando corresponden a un proyecto totalmente elaborado y susceptible de ser ejecutado.
- *Acciones ficticias*: si corresponden a un proyecto idealizado, elaborado de modo incompleto o construido en la imaginación.
- *Acciones realistas*: cuando corresponden a un proyecto cuya ejecución puede ser tomada en cuenta.
- *Acciones no realistas*: aunque constituyen un buen soporte para la discusión y el análisis, corresponden a objetivos incompatibles.

- *Acción potencial*: es la acción real o ficticia, provisionalmente juzgada realista al menos por un actor o presumida como tal por el hombre de estudio. El conjunto de acciones potenciales lo representamos por A . Este conjunto A se considera una cantidad finita y podemos escribirlo enumerando las acciones que lo componen $A = \{a, b, c, \dots\}$.

Según Vincke [203], un conjunto de acciones A , puede ser:

- Estable, no es susceptible de ser cambiado en el proceso.
- Evolutivo, cuando puede haber cambios a lo largo del proceso.

1.6 Las diferentes problemáticas

Definido y delimitado el conjunto de acciones potenciales A , interesaría saber qué es lo que se pretende, es decir cómo convendría modelar la situación. Según Roy [161][163], deben considerarse las siguientes problemáticas de referencia:

- *Problemática P_α* (o problemática de selección): el decisor desea seleccionar una acción de A . La investigación se orienta a proporcionar la mejor solución posible.
- *Problemática P_γ* (o problemática de ordenación): el decisor desea una ordenación de todas las acciones de A , por ejemplo de la mejor a la menos buena. La investigación se orienta a obtener una ordenación (un preorden total) sobre A .
- *Problemática P_β* (o problemática de clasificación): el decisor desea afectar cada acción a una categoría definida a priori. El procedimiento debe proceder a esta clasificación.

1.7 Modelado de las preferencias

El modelado de las preferencias del decisor en relación a las acciones, juega un papel fundamental en el proceso de ayuda a la decisión. Su omisión puede llevar a elaborar modelos inadecuados de la realidad que se pretende estudiar. Obras fundamentales [62][110][120], publicadas en los años 1970, dan cuerpo a lo que se llama *la teoría clásica del modelado de preferencias*. Esta teoría se caracteriza por la utilización esquemática de un lenguaje particular acompañado de una sintaxis específica. En esta sección se presentan los principales conceptos utilizados en el modelado de las preferencias. Primero se introduce la noción de relación binaria, que es la herramienta central en la mayor parte de los modelos de preferencias. Se define a continuación

lo que se entiende por sistema relacional de preferencias. Finalmente, se presentan algunas de las principales estructuras clásicas de preferencias.

1.7.1 Relación binaria

Definición

Se llama relación binaria R en un conjunto A , a todo subconjunto del producto cartesiano $A \times A$. Si una pareja (a, b) está en R , se escribirá $(a, b) \in R$ o aRb . En caso contrario se notará $(a, b) \notin R$ o $a \neg Rb$.

Propiedades

Se dice que una relación binaria R en A es (ver para más detalles [14][47]):

- Reflexiva: $\forall a \in R, aRa$
- Irreflexiva: $\forall a \in R, a \neg Ra$
- Simétrica: $\forall a, b \in R, aRb \Rightarrow bRa$
- Asimétrica: $\forall a, b \in R, aRb \Rightarrow b \neg Ra$
- Antisimétrica: $\forall a, b \in R, aRb \text{ y } bRa \Rightarrow a=b$
- Transitiva: $\forall a, b, c \in R, [aRb \text{ y } bRc] \Rightarrow aRc$
- Negativamente transitiva: $\forall a, b, c \in R, [a \neg Rb \text{ y } b \neg Rc \Rightarrow a \neg Rc]$
- Semi-transitiva o cuasi-transitiva: $\forall a, b, c, d \in R, [aRb \text{ y } bRc] \Rightarrow [aRd \text{ o } dRc]$
- Ferrers: $\forall a, b, c, d \in R, [aRb \text{ y } cRd] \Rightarrow [aRd \text{ o } cRb]$
- Comparable o completa: $\forall a, b \in R, [aRb \text{ o } bRa \text{ o ambas a la vez}]$
- Conexa: $\forall a, b \in R, \text{ si } a \neq b \Rightarrow [aRb \text{ o } bRa \text{ o ambas a la vez}]$
- Sean R_1, R_2, \dots, R_n , n relaciones binarias sobre el conjunto A . Se dice que esas n relaciones son mutuamente exclusivas si: $\forall a, b \in R, \forall i \in \{1, 2, \dots, n\}, aR_i b \Rightarrow [(a \neg R_j b) \text{ y } (b \neg R_j a), \forall j \neq i]$
- Sean R_1, R_2, \dots, R_n , n relaciones binarias sobre el conjunto A . Se dice que esas n relaciones son exhaustivas si: $\forall a, b \in R, \exists i \in \{1, 2, \dots, n\} \text{ tal que } aR_i b \text{ o } bR_i a$
- Toda relación asimétrica es irreflexiva
- Una relación es asimétrica si y solo si es irreflexiva y antisimétrica

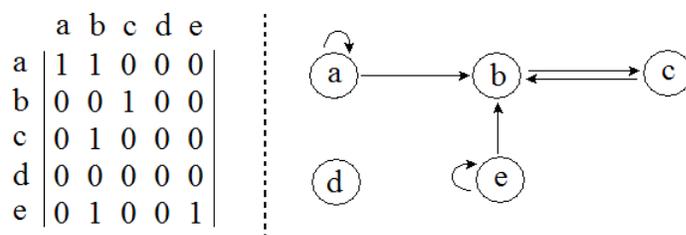


Figura 1.2: Representación matricial de la relación binaria R sobre el conjunto A (izquierda) y del grafo (A,R) (derecha).

- Una relación es completa si y solo si es conexa y reflexiva
- Una relación asimétrica y negativamente transitiva es transitiva
- Una relación completa y transitiva es negativamente transitiva
- Dada una relación binaria R sobre el conjunto A se define:
 - Su parte simétrica I_R tal que: $\forall a, b \in R, aI_Rb$ si $[aRb$ y $bRa]$
 - Su parte asimétrica P_R tal que: $\forall a, b \in R, aP_Rb$ si $[aRb$ y $b \neg Ra]$

Representación matricial

Una relación binaria R en A , se puede representar de forma matricial de manera que el elemento de la fila i y columna j es un 1 si a_iRa_j o un 0 si $a_i \neg Ra_j$. Con esta representación, la reflexividad significa la presencia de un 1 sobre la diagonal principal de la matriz.

Representación gráfica

Otra forma de representar una relación binaria R en A , es mediante un grafo (A,R) donde A es un conjunto de nodos y R un conjunto de arcos, de manera que si aRb entonces existe un arco que va de a hasta b . Con esta representación la reflexividad se traduce por la presencia de un bucle en cada nodo. La simetría significa que la presencia de un arco orientado de a hacia b implica la existencia de un arco de b hacia a .

Ejemplo

Supóngase el conjunto $A=\{a,b,c,d,e\}$ y considerese sobre A la relación binaria $R=\{(a,a)(a,b)(b,c)(c,b)(e,b)(e,e)\}$. La figura 1.2 muestra la representación matricial de R sobre A (izquierda) y el grafo (A,R) (derecha).

1.7.2 Sistema relacional de preferencias

Supongase un decisor de un proceso de decisión, al que se pregunta sobre un par de acciones (a,b) de un conjunto A , si “la acción a es al menos tan buena como la acción b ”. Si suponemos que las únicas respuestas posibles son “sí” o “no”, plantear dicha cuestión a todas las parejas de acciones de A , lleva a definir una relación binaria E sobre A , de forma que aEb si y solo si la respuesta a la pregunta es “sí”.

Supóngase ahora, que a la cuestión antes planteada se autorizan además de las respuestas “sí” o “no”, respuestas del tipo “la acción a es indiferente a la acción b ”, “la acción a es incomparable con la acción b ”, “la acción a es fuertemente (débilmente) preferida a la acción b ”. Admitir tales respuestas, implica utilizar un lenguaje (más amplio) que permite definir una relación binaria E sobre A de forma que para todo par de acciones (a,b) de A , un decisor estará confrontado a una y sola una de las situaciones fundamentales siguientes:

- La acción a la prefiere sin ningún género de dudas a la acción b [aEb y $b\neg Ea$]. Se habla entonces de **Preferencia Estricta aPb** . P es una relación binaria sobre A asimétrica e irreflexiva y por convención se utiliza la representación gráfica de la figura 1.3 izquierda.
- Es indiferente entre las dos acciones [aEb y bEa]. Se habla de **Indiferencia aIb** . I es una relación binaria sobre A reflexiva y simétrica. Por convención I se representa según la figura 1.3 centro.
- No sabe si la acción a la prefiere estrictamente a la acción b o si es indiferente entre las dos y en consecuencia no puede diferenciar entre la preferencia estricta y la indiferencia [$(aEb$ y $b\neg Ea)$ o $(aEb$ y $bEa)$]. Se habla entonces de **Preferencia Débil aQb** . Q es una relación binaria sobre A asimétrica e irreflexiva.
- No las puede comparar [$a\neg Eb$ y $b\neg Ea$]. Se habla entonces de **Incomparabilidad aJb** . Por convenio se usa la representación gráfica de la figura 1.3 derecha.

Estas cuatro situaciones básicas, sirven para modelar las preferencias del decisor haciendo corresponder a cada par de acciones de A , bien una sola de ellas, bien una agrupación de las cuatro. Entre las agrupaciones con un interés particular tenemos:

- **No Preferencia (\approx)**: no hay razones claras a favor de una Preferencia Estricta o Débil, por tanto, agrupa sin posibilidad de diferenciarlas, la indiferencia y la incomparabilidad $a \approx b \Leftrightarrow (aIb \text{ o } aJb)$.

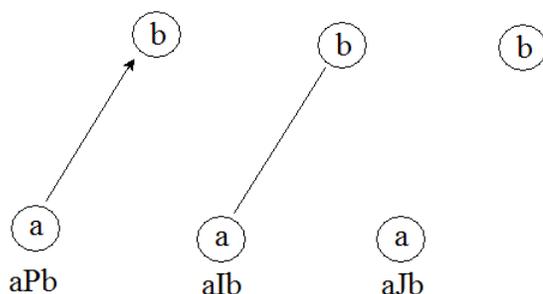


Figura 1.3: Convenciones gráficas para la preferencia P, la indiferencia I y la incomparabilidad J.

- **Preferencia en sentido amplio (\gg):** hay preferencia de una de las acciones sobre la otra, pero la Preferencia puede ser Estricta o Débil. $a \gg b \Leftrightarrow (aPb \text{ o } aQb)$.
- **Presunción de preferencia (L):** hay razones claras que justifican la Preferencia Débil e incluso la Indiferencia, pero sin establecer una separación significativa entre las situaciones de Preferencia Débil y de Indiferencia. De cualquier forma, está bien claro que no hay Incomparabilidad ni Preferencia Fuerte. $aLb \Leftrightarrow (aQb \text{ o } aIb)$.
- **Superación (S):** se puede justificar una Preferencia Fuerte o Débil o incluso la Indiferencia. $aSb \Leftrightarrow (aPb \text{ o } aQb \text{ o } aIb)$.

Para Roy y Bouyssou [165], estas ocho relaciones binarias P, I, Q, J, \approx , \gg , L y S constituyen un Sistema Relacional de Preferencia sobre el conjunto A si:

1. Pueden ser tomadas como representación de las preferencias de un decisor con respecto a las acciones del conjunto A.
2. Son exhaustivas y mutuamente excluyentes.

1.7.3 Estructuras de preferencias clásicas

Estructura de preorden total (o completo)

Una estructura de preferencia (T, V) , es una estructura de preorden total (algunos autores lo representan por \succeq) sobre un conjunto A [47] si:

- La incomparabilidad está vacía ($J=0$).

- T y V son exhaustivas y mutuamente exclusivas.
- T es simétrica y transitiva.
- V es asimétrica y transitiva.

Un preorden total (T,V) sobre un conjunto A , corresponde a la situación en la que se pueden ordenar todos los elementos de A (de mejor a peor) y se permite el exaequo. En problemas reales, un preorden total (T,V) puede ser representado por una función g de valores reales definida sobre A de forma que $\forall a, b \in A$:

$$\begin{cases} aVb \Leftrightarrow g(a) > g(b) \\ aTb \Leftrightarrow g(a) = g(b) \end{cases} \quad (1.7.1)$$

Si suponemos la transitividad de I (o \approx) y de P (o \gg), entonces los sistemas de preferencias (I,P) , (\approx,P) , (\approx,\gg) o (I,\gg) tienen estructura de preorden total.

Problema clásico

El preorden total (I,P) es la estructura de preferencia utilizada en todos los problemas de optimización clásica y puede ser representada por la función g de valores reales definida sobre A de forma que $\forall a, b \in A$:

$$\begin{cases} aPb \Leftrightarrow g(a) > g(b) \\ aIb \Leftrightarrow g(a) = g(b) \end{cases} \quad (1.7.2)$$

La función g es aquella que es preciso maximizar o minimizar, y se denomina según el contexto función de utilidad, función valor, función objetivo, criterio, etc.

Ejemplo

Dado el conjunto de acciones $A=\{a,b,c,d\}$, se puede verificar el preorden total dado por $(I,P)=\{(a,a)(a,b)(a,c)(a,d)(b,b)(b,c)(b,d)(c,b)(c,c)(c,d)(d,d)\}$ donde la relación de indiferencia está dada por $I=\{(a,a)(b,b)(b,c)(c,b)(c,c)(d,d)\}$ y la de preferencia por $P=\{(a,b)(a,c)(a,d)(b,d)(c,d)\}$, considerando la representación matricial (figura 1.4 izquierda) y gráfica (figura 1.4 derecha sin bucles de reflexividad). El preorden total establecido corresponde a: $a \succ b \sim c \succ d$.

Estructura de orden total

Una estructura de orden total (algunos autores lo representan por \succ) sobre un conjunto A , es un caso particular de la estructura preorden completo, de forma que

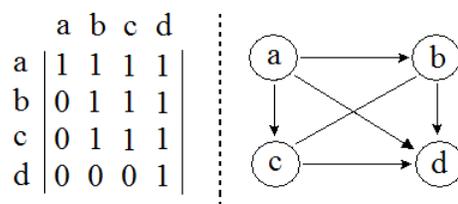


Figura 1.4: Representación del Preorden total (I,P): matricial (izquierda) y gráfica (derecha).

la relación T no se verifica más que entre acciones idénticas. Correspondería a la situación en la que se pueden ordenar todos los elementos de A (de mejor a peor) sin posibilidad de exaequos. En problemas reales, el orden completo (T,V) puede ser representado por una función g de valores reales definida sobre A de forma que $\forall a, b \in A$:

$$\begin{cases} aVb \Leftrightarrow g(a) > g(b) \\ aTb \Leftrightarrow g(a) = g(b) \\ g(a) = g(b) \Rightarrow a \equiv b \end{cases} \quad (1.7.3)$$

Ejemplo

Dado el conjunto de acciones $A=\{a,b,c,d\}$, se puede verificar el orden completo $(I,P)=\{(a,a)(a,b)(a,c)(a,d)(b,b)(b,c)(b,d)(c,c)(c,d)(d,d)\}$ donde la relación de indiferencia está dada por $I=\{(a,a)(b,b)(c,c)(d,d)\}$ y la relación de preferencia por $P=\{(a,b)(a,c)(a,d)(b,c)(b,d)(c,d)\}$, considerando la representación matricial (figura 1.5 izquierda) y gráfo (figura 1.5 derecha sin bucles de reflexividad). El orden total establecido corresponde a: $a \succ b \succ c \succ d$.

Estructuras parciales

Las estructuras vistas hasta ahora son completas. Sin embargo hay situaciones donde puede ser necesario hacer aparecer la incomparabilidad de las acciones en el modelado de las preferencias (consultar a Roy [163] y Flament [63]). Se estudian a continuación dos estructuras que admiten la incomparabilidad: el preorden parcial y el orden parcial.

Estructura de preorden parcial

Una estructura de preferencia (T,V,W) es una estructura de preorden parcial sobre un conjunto A [47] si:

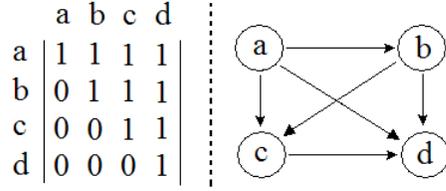


Figura 1.5: Representación del Orden total (I,P): matricial (izquierda) y gráfica (derecha).

- T, V y W son exhaustivas y mutuamente exclusivas.
- W es simétrica e irreflexiva.
- V es asimétrica.
- T es simétrica y reflexiva.
- (TUV) es transitiva.

La estructura de preorden parcial es una generalización de la estructura de preorden completo admitiendo la incomparabilidad J y conservando la propiedad de la transitividad.

En problemas reales, un preorden parcial (T,V,W) puede ser representado por una función g de valores reales definida sobre A de forma que $\forall a, b \in A$:

$$\begin{cases} aVb \Rightarrow g(a) > g(b) \\ aTb \Rightarrow g(a) = g(b) \end{cases} \tag{1.7.4}$$

Si se supone la transitividad de P, \approx , I o \gg , entonces los sistemas de preferencias (I,P,J), (I, \approx ,J), (\gg ,P,J) y (\gg , \approx ,J) tienen estructura de preorden parcial.

Problema clásico

En un problema de optimización multiobjetivo, el preorden parcial (I,P,J) es la estructura de preferencia utilizada en la relación de dominancia debil de Pareto (ver la figura 1.6 derecha), y puede ser representada por la función g de valores reales definida sobre A de forma que $\forall a, b \in A$:

$$\begin{cases} aPb \Rightarrow g(a) > g(b) \\ aIb \Rightarrow g(a) = g(b) \end{cases} \tag{1.7.5}$$

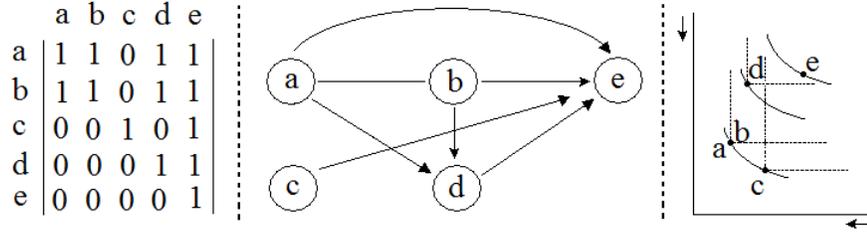


Figura 1.6: Preorden parcial (I,P,J): representación matricial (izquierda), gráfica (centro) y dominancia debil de Pareto (derecha).

Ejemplo

Dado el conjunto de acciones $A=\{a,b,c,d,e\}$, se puede verificar el preorden parcial $(I,P,J)=\{(a,a)(a,b)(a,d)(a,e)(b,b)(b,a)(b,d)(b,e)(c,c)(c,e)(d,d)(d,e)(e,e)\}$, donde $I=\{(a,a)(a,b)(b,b)(b,a)(c,c)(d,d)(e,e)\}$, $P=\{(a,d)(a,e)(b,d)(b,e)(c,e)(d,e)\}$ y la incomparabilidad $J=\{(a,c)(c,a)(b,c)(c,b)(c,d)(d,c)\}$, considerando la representación matricial (figura 1.6 izquierda) y el gráfico (figura 1.6 centro sin bucles de reflexividad).

Estructura de orden parcial

Una estructura de orden parcial resulta un caso particular de la estructura preorden parcial, imponiendo que $\forall a,b \in A, aTb \Leftrightarrow a = b$, de manera que la relación T no se verifica más que entre acciones idénticas. Una estructura de orden parcial (T,V,W) sobre un conjunto A , puede en un problema real, ser representada por una función g de valores reales sobre A de forma que $\forall a,b \in A$:

$$\begin{cases} aVb \Rightarrow g(a) > g(b) \\ aTb \Rightarrow g(a) = g(b) \\ g(a) = g(b) \Rightarrow a \equiv b \end{cases} \quad (1.7.6)$$

Problema clásico

El orden parcial (I,P,J) es la estructura de preferencia utilizada en la relación de dominancia de Pareto (ver figura 1.7 derecha) de un problema de optimización multiobjetivo, y puede ser representada por la función g de valores reales definida sobre A de forma que $\forall a,b \in A$:

$$\begin{cases} aPb \Rightarrow g(a) > g(b) \\ aIb \Rightarrow g(a) = g(b) \\ g(a) = g(b) \Rightarrow a \equiv b \end{cases} \quad (1.7.7)$$

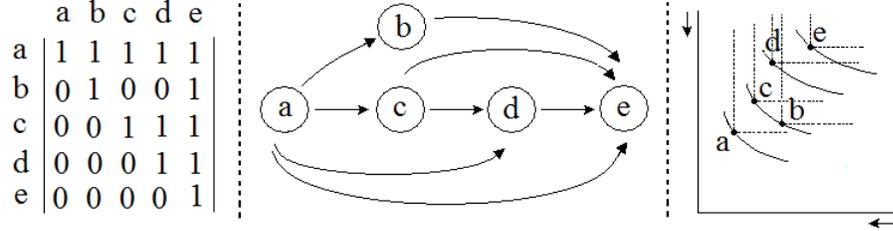


Figura 1.7: Orden parcial (I,P,J): representación matricial (izquierda), gráfica (centro) y dominancia de Pareto (derecha).

Ejemplo

Dado el conjunto de acciones $A=\{a,b,c,d,e\}$, se puede verificar el orden parcial dado por $(I,P,J)=\{(a,a)(a,b)(a,c)(a,d)(a,e)(b,b)(b,e)(c,c)(c,d)(c,e)(d,d)(d,e)(e,e)\}$, considerando la representación matricial (figura 1.7 izquierda) y gráfica (figura 1.7 centro sin bucles de reflexividad) y donde se tiene que $I=\{(a,a)(b,b)(c,c)(d,d)(e,e)\}$, $P=\{(a,b)(a,c)(a,d)(a,e)(b,e)(c,d)(c,e)(d,e)\}$ y $J=\{(b,c)(c,b)(b,d)(d,b)\}$.

La transitividad y las estructuras de orden de intervalo, cuasi-orden y pseudo-orden

Las estructuras vistas hasta ahora, suponen la transitividad de P, \approx, I o \gg . Sin embargo, esta hipótesis ha sido criticada por numerosos autores (ver por ejemplo a Luce [127]), por suponer una discriminación perfecta entre dos acciones con valores próximos pero distintos, lo que conduce a un modelado poco realista de las preferencias. Esto justifica la introducción de las estructuras de orden de intervalo y cuasi-orden (ambas no imponen la transitividad a I o \approx), y la estructura de pseudo-orden (no impone la transitividad a las cuatro relaciones P, \approx, I o \gg). Para un mayor detalle del problema de la transitividad y las estructuras de orden de intervalo, cuasi-orden y pseudo-orden, consultar [9][14][47][153].

1.8 Modelado de las consecuencias

Para poder comparar las distintas acciones en términos de preferencia, indiferencia, incomparabilidad,... es preciso que conozcamos cuáles son las consecuencias de la aplicación de cada una de las acciones de A . Las consecuencias pueden ser múltiples o no conocerse bien e incluso pueden estar mal delimitadas. Al conjunto de las consecuencias de la acción a , se le denomina nube de consecuencias $V(a)$.

Una vez que la nube de consecuencias es conocida, es necesario identificar y aislar

las consecuencias elementales que la integran. Cada consecuencia elemental debe estar bien identificada para que todos los actores comprendan su significado, y además debe poder ser descrita por aquello en que se manifiesta cuando se ejecuta cada acción (estados asociados). Para el ejemplo de compra de un automóvil, una consecuencia elemental podría ser el precio, y el valor exacto para cada coche sería el estado asociado.

Para Roy [163], cuando los estados asociados a una consecuencia están ordenados según un preorden total del tipo \succeq , se dice entonces que los estados forman una *Escala de Preferencia E*. Asimismo cuando se comparan dos acciones a y b, que equivale a comparar los dos estados asociados a ambas e_a y e_b , cualquier actor admite que:

- hay indiferencia aIb cuando e_a y e_b son iguales.
- hay preferencia aPb cuando $e_a > e_b$ en el preorden (e_b está después de e_a).

Una consecuencia elemental, que tiene todos sus estados asociados ordenados según un preorden completo formando una escala E, se llama *Dimensión*. Por tanto, el precio de un automóvil, ordenado de modo que el precio más pequeño sea el primero (el actor prefiere un precio bajo) y el mayor el último, es una *Dimensión*.

Para cada dimensión i, la regla o técnica que nos sirve para conocer los niveles que contendrá la escala E_i si se ejecuta la acción a, se denomina *Indicador de Estado* γ_i . El indicador de estado se dice puntual si al ejecutar la acción a, $\gamma_i(a)$ se reduce a un solo nivel de E_i , se define no puntual en caso contrario.

Se denomina indicador de dispersión $\delta_i^a(e)$, a aquella función definida sobre una parte de la escala E_i (propia de la acción a) que cumple:

1. $e \in \gamma_i(a) \Rightarrow \delta_i^a(e) > 0$.
2. $\delta_i^a(e_a) = \delta_i^b(e_b) \Leftrightarrow$ para evaluar las consecuencias de a y b sobre la dimensión i, el hombre de estudio acuerda una importancia o verosimilitud comparables a los niveles e_a y e_b .
3. $\delta_i^a(e_a) < \delta_i^b(e_b) \Leftrightarrow$ para evaluar las consecuencias de a y b sobre la dimensión i, el hombre de estudio acuerda una importancia o verosimilitud menor al nivel e_a que al nivel e_b .

En resumen, las consecuencias de cada una de las acciones de A serán analizadas en relación a un conjunto de n dimensiones (consecuencias elementales cuyos estados siguen un preorden completo y forman una escala) cumpliéndose que:

- Cada dimensión, está bien identificada en su contenido y en su significado.

- A cada dimensión i , se le puede asociar una escala E_i cuyos niveles siguen un preorden completo.
- Es posible asociar a cada dimensión i , un indicador de estado γ_i y un estado de dispersión δ_i^a .

De esta forma, las consecuencias de la puesta en práctica de una acción potencial, pueden ser representadas por: $\tau(a) = [(\gamma_1(a), \delta_1^a), (\gamma_2(a), \delta_2^a), \dots, (\gamma_n(a), \delta_n^a)]$. El conjunto de las representaciones que corresponden a las acciones de A , lo representamos por $\tau(A)$.

1.9 Los criterios

Para un actor implicado en un proceso de decisión, una representación $\tau(A)$ de las consecuencias de las acciones, analizada en relación a un conjunto de n dimensiones, puede no ser estable ni estar definido de forma rigurosa. En este sentido, un criterio trata de resumir con ayuda de una función, las evaluaciones de una acción sobre diversas dimensiones que se asocian al mismo eje de significación o punto de vista.

Según Roy y Bouyssou [165], un *criterio* g es un modelo a partir del cual, se reconoce poder fundar la proposición $g(a) \geq g(b) \Rightarrow aS_g b$, donde S_g es una relación de superación con el contenido semántico *al menos tan bueno como*, relativamente a las evaluaciones sobre las únicas dimensiones tomadas en cuenta en la definición del criterio g .

1.10 La agregación

Definidas las acciones y criterios que intervienen en el proceso de toma de decisiones, se debe a continuación obtener una serie de evaluaciones para cada acción según cada criterio. Se trata ahora de saber en qué nos basamos para poder decir que una acción b es al menos tan buena como otra a teniendo en cuenta, no ya un único criterio, sino todos los criterios. Es decir, ¿cómo logramos agregar las evaluaciones?

Los métodos de agregación son numerosos, esto es debido según Schärling [181], a que en la naturaleza de las cosas, ningún método respeta la totalidad de las exigencias que un actor podría encontrar normales en la idea del multicriterio, siendo preciso decidir en qué exigencia se va a ceder. Para Roy y Bouyssou [165], un método de agregación multicriterio, es una regla o procedimiento, que permite establecer sobre el conjunto de evaluaciones de las acciones según cada uno de los criterios y sobre las informaciones entre criterios (coeficientes de importancia, tasas de sustitución, umbrales de veto, etc.), uno o varios sistemas relacionales de preferencias sobre el

conjunto A de las acciones, recordemos que un sistema relacional de preferencias puede tomar diversas formas (I,P), (I,P,J), (S,J),... Roy [1985] distingue tres enfoques o métodos diferentes de agregación: métodos de agregación completa, métodos de agregación parcial y métodos de agregación local interactivos.

1.10.1 Métodos de agregación completa

Estos métodos se caracterizan por agregar los diferentes puntos de vista o criterios (eventuales pesos les pueden ser atribuidos) en una única función la cual debe ser posteriormente maximizada o minimizada. Roy [163] y Maystre [131] los denominan *enfoque del criterio único de síntesis eliminando toda incomparabilidad*.

Con esta técnica de agregación, todas las soluciones son comparables (no se admite la incomparabilidad), consiguiéndose una ordenación completa para cualquier par de alternativas a y b , es decir, aPb , bPa o aIb ; ello permite tratar fácilmente tanto las problemáticas de selección P_α como las problemáticas de ordenación P_γ . La indiferencia I y la preferencia P son transitivas, y el sistema relacional de preferencias es un preorden completo que toma la forma (I,P).

En el dominio de la agregación completa, existen un gran número de métodos. Se destacan entre otros, la suma ponderada [9][153][197], TOPSIS⁴ [101], MAUT⁵ [62][110], UTA⁶ [106], SMART⁷ [45] y AHP⁸ [169][170]. Seguidamente se presentan en detalle los métodos de la suma ponderada y el método TOPSIS.

Suma ponderada

El método de la suma ponderada es uno de los procedimientos de agregación completa más conocido y utilizado en AMD. El mecanismo de funcionamiento es el siguiente:

1. Se elabora la matriz de decisión (tabla 1.2) del problema, de forma que un conjunto n de acciones o alternativas $A=(a_j, j=1,2,\dots,n)$ es comparado con respecto a un conjunto k de criterios $G=(g_i, i=1,2,\dots,k)$. La utilidad $g_i(a_j)$, que para un decisor tiene la alternativa j , respecto al criterio i , se recoge en la evaluación $x_{ji}=g_i(a_j)$. Cada valor x_{ji} procede de la construcción de una verdadera función de utilidad o bien de una evaluación natural como por ejemplo el precio.
2. Para un decisor, no todos los criterios tienen porqué tener la misma importancia. Mediante pesos (o ponderaciones) w_i , (ver tabla 1.2) el decisor, acorde a sus

⁴Technique for Order Preference by Similarity to an Ideal Solution

⁵Multi-Attribute Utility Theory

⁶Utilités Additives

⁷Similarity Measured Anchored Ranking Technique

⁸Analytic Hierarchy Process

Tabla 1.2: Matriz de decisión para un conjunto A de acciones y un conjunto G de criterios.

	g ₁	g ₂	·	·	g _i	·	·	g _k	
a ₁	x ₁₁	x ₁₂	·	·	x _{1i}	·	·	x _{1k}	R(a ₁)
a ₂	x ₂₁	x ₂₂	·	·	x _{2i}	·	·	x _{2k}	R(a ₂)
·	·	·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·	·	·
a _j	x _{j1}	x _{j2}	·	·	x _{ji}	·	·	x _{jk}	R(a _j)
·	·	·	·	·	·	·	·	·	·
·	·	·	·	·	·	·	·	·	·
a _n	x _{n1}	x _{n2}	·	·	x _{ni}	·	·	x _{nk}	R(a _n)
Pesos	w ₁	w ₂	·	·	·	·	·	w _k	

preferencias cuantifica los criterios de manera que $\sum_{i=1}^k w_i = 1$ con $w_i \geq 0$ para

todo i . Un peso de valor cero en algún criterio, permite la eliminación de dicho criterio.

3. Si las escalas de los criterios son diferentes, se normalizan (ver [9][153]) las evaluaciones x_{ji} de la matriz de decisión y se obtienen los valores v_{ji} .
4. Para cada acción a_j se determina su evaluación global (ver tabla 1.2) de forma:

$$R(a_j) = \sum_{i=1}^k w_i v_{ji}$$

5. El método concluye que la acción a_1 es preferida a la acción a_2 si:

$$R(a_1) > R(a_2)$$

en caso de empate entre varias acciones, se prefiere cualquiera de ellas.

Ejemplo

Como ejemplo se ha utilizado el problema de Vallin y Vanderpooten [197]. Las utilidades de las acciones del problema con respecto a los criterios g_1 y g_2 , se muestran en la matriz de decisión dada por la tabla 1.3, y donde las escalas que se manejan son iguales para ambos criterios. La representación de las acciones en el espacio de los

Tabla 1.3: Matriz de decisión del problema de Vallin y Vanderpooten.

Acción	max. g_1	max. g_2
a	18	4
b	10	10
c	5	18
d	9	7
Pesos	0.5	0.5

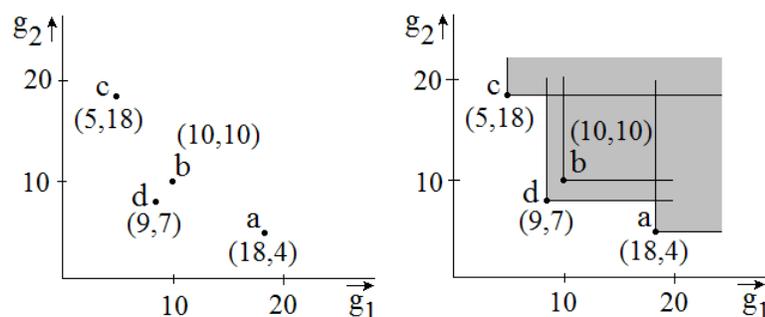


Figura 1.8: Representación de las acciones del problema de Vallin y Vanderpooten (izquierda) y test de dominancia (derecha).

criterios, se muestra en la figura 1.8 izquierda. El resultado del test de dominancia se ilustra en la figura 1.8 derecha. El método de la suma ponderada determina para cada acción la siguiente evaluación global:

la mejor evaluación resulta para la acción c, la cual es elegida por el centro decisor.

$$R(a) = 0.5 * 18 + 0.5 * 4 = 11$$

$$R(b) = 0.5 * 10 + 0.5 * 10 = 10$$

$$R(c) = 0.5 * 5 + 0.5 * 18 = 11.5$$

$$R(d) = 0.5 * 9 + 0.5 * 7 = 8$$

Discusión

El método de la suma ponderada, es utilizado abundantemente en diversos contextos pues es muy intuitivo y sencillo de utilizar, aunque presenta ciertas limitaciones que se analizan a continuación:

1. No hay una correspondencia intuitiva entre valores de los pesos y la acción óptima propuesta [197]. En el ejemplo, el decisor decidió unos pesos idénticos a

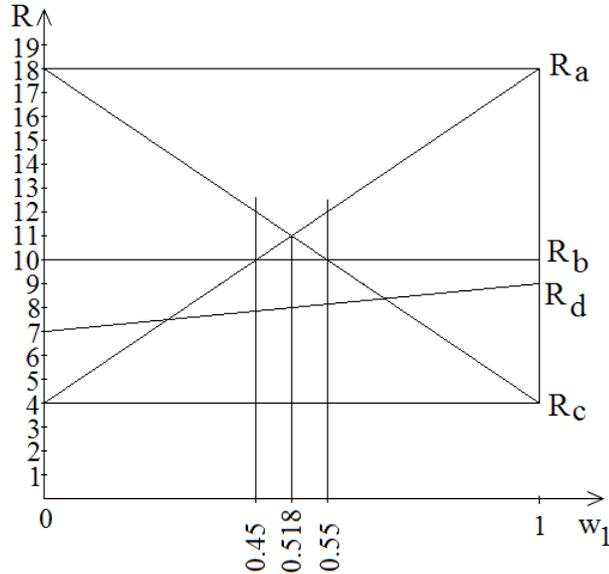


Figura 1.9: Evolución de la suma ponderada en función del valor w_1 , para las cuatro acciones del problema de Vallin y Vanderpooten.

cada criterio $w_1 = w_2 = 0.5$, esto podría hacer pensar al decisor, que se va a dar preferencia a una acción bien equilibrada en ambos criterios, y esto no ocurre precisamente aquí, pues se privilegia una acción muy desequilibrada como es la c, (b es la única acción equilibrada en el ejemplo).

2. Existe una extrema sensibilidad del resultado a variaciones ínfimas de los valores de los pesos [197]. Si en el ejemplo, expresamos para cada acción, el valor de la suma ponderada en función del peso w_1 tenemos:

$$\begin{aligned} R(a) &= 18w_1 + 4w_2 = 14w_1 + 4 \\ R(b) &= 10w_1 + 10w_2 = 10 \\ R(c) &= 5w_1 + 18w_2 = -13w_1 + 18 \\ R(d) &= 9w_1 + 7w_2 = 2w_1 + 7 \end{aligned}$$

La representación gráfica, de la evolución del valor R según el valor de w_1 , para cada una de las cuatro acciones, se muestra en la figura 1.9. Se visualiza, que en un pequeño intervalo $[0.45, 0.55]$, a y c son acciones óptimas. En concreto la acción c es óptima en el intervalo $w_1 \in [0.45, 0.518]$, mientras que la acción a es óptima en $w_1 \in [0.518, 0.55]$.

3. Ciertas acciones no dominadas fuera de una combinación convexa, no pueden

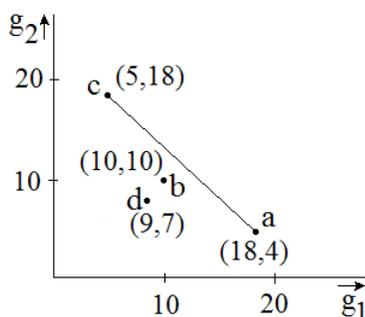


Figura 1.10: Acción b fuera de la combinación convexa a-c en el problema de Vallin y Vanderpooten.

ser obtenidas como soluciones óptimas [197]. Supongase en el ejemplo, que se quiere encontrar un juego de pesos tal que haga aparecer la acción b como óptima. Vemos en la figura 1.9 que esto no es posible, pues para $w_1 \in [0.00, 0.518]$ la acción c es la óptima, sin embargo para $w_1 \in [0.518, 1]$ es la acción a la óptima. Otro punto de vista de ver esta situación, es a través de la convexidad. En efecto, b es una acción no dominada, pero está situada debajo de la combinación convexa de los modelos a y c (ver la figura 1.10).

4. El método de la suma ponderada supone la existencia de una función de utilidad cardinal aditiva para los criterios, lo que a su vez presupone independencia entre los criterios, y comparabilidad entre los criterios de los valores de las acciones [9][153].

Método TOPSIS

El método TOPSIS, fue propuesto por Hwang y Yoon [101] para la resolución de problemas de ayuda multicriterio a la decisión (problemas con un número finito de soluciones), y está fundado en el axioma de Zeleny [218]: *es racional elegir una acción lo más próxima a la ideal o lo más alejada de la anti-ideal*. En la situación más general (figura 1.11 izquierda), TOPSIS establece que se debe elegir aquella acción (a_j) con menor distancia a la acción ideal (I^+) y mayor distancia a la acción anti-ideal (I^-).

Sin embargo la figura 1.11 (derecha), muestra que la acción a_z es la más cercana a la acción ideal, mientras que la acción a_j es la más alejada de la acción anti-ideal, aquí, TOPSIS afronta el dilema de trabajar con la acción ideal o la anti-ideal. TOPSIS resuelve el dilema, utilizando el concepto de *similaridad al ideal* [45].

En cuanto a las distancias, estas se calculan para un valor particular de la métrica

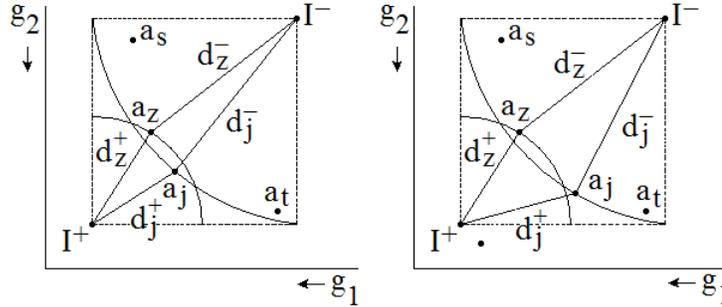


Figura 1.11: Situación general en el método TOPSIS, con a_j como mejor acción (izquierda) y dilema de trabajar con la solución a_j o a_z (derecha).

p (con: $1 \leq p \leq \infty$) en la ecuación de la función distancia:

$$L_p = \left[\sum_{i=1}^k w_i^p |v_{ji} - I^+|^p \right]^{\frac{1}{p}} \quad (1.10.1)$$

donde v_{ji} es el valor normalizado de la acción a_j implementada para el criterio g_i , I^+ es el vector cuyas coordenadas corresponde al punto de referencia (con TOPSIS este punto corresponde a la acción ideal o la anti-ideal) y w_i es el vector de pesos acorde

a las preferencias del decisor con $\sum_{i=1}^k w_i = 1$ y $w_i \geq 0$ para todo i .

El método se compone de un procedimiento de 6 etapas:

1. Se construye la matriz de decisión (tabla 1.2) del problema. Las evaluaciones $x_{ji} = g_i(a_j)$, como en el método de las ponderaciones, corresponde a la utilidad $g_i(a_j)$, que para el decisor tiene la alternativa j , respecto al criterio i .
2. El decisor asigna a los criterios un peso $w_i \geq 0$ según sus preferencias.
3. Se normalizan las evaluaciones x_{ji} y se obtienen los valores v_{ji} .
4. Se determina la acción ideal $I^+ = (v_1^+, v_2^+, \dots, v_k^+)$, construida con las mejores acciones de cada criterio y la acción anti-ideal $I^- = (v_1^-, v_2^-, \dots, v_k^-)$ constituida por las peores acciones de cada criterio (figura 1.12).

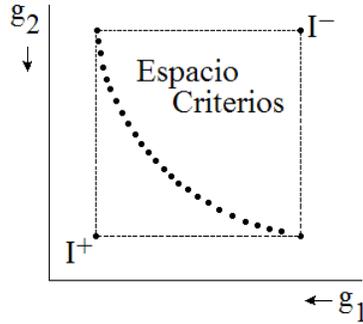


Figura 1.12: Acciones ideal (I^+) y anti-ideal (I^-).

5. Se calculan los valores de las distancias Euclídeas de cada acción respecto la acción ideal según:

$$d_j^+ = \sqrt{\sum_{i=1}^k w_i (v_{ji} - I^+)^2} \quad j = 1, 2, \dots, n \quad (1.10.2)$$

De forma similar se obtienen los valores de las distancias Euclídeas de cada acción respecto la acción anti-ideal:

$$d_j^- = \sqrt{\sum_{i=1}^k w_i (v_{ji} - I^-)^2} \quad j = 1, 2, \dots, n \quad (1.10.3)$$

6. Se determina el ratio de similaridad D_j^+ o cercanía relativa a la acción ideal a partir de la ecuación:

$$D_j^+ = \frac{d_j^-}{d_j^+ + d_j^-} \quad \text{con } D_j^+ = [0, 1] \quad (1.10.4)$$

7. Se clasifican las acciones según el orden de preferencia: de mayor a menor similaridad D_j^+ .

Tabla 1.4: Datos normalizados, acciones ideal y anti-ideal en el problema de Vallin y Vanderpooten.

Acción	max. g ₁	max. g ₂
a	0.4286	0.1026
b	0.2381	0.2564
c	0.1190	0.4615
d	0.2143	0.1795
Pesos	0.5	0.5
ideal	0.4286	0.4615
anti-ideal	0.1190	0.1026

Ejemplo

Aplicando el método TOPSIS al problema de Vallin y Vanderpooten [197] presentado en la sección 1.10.1 (ahora se ha considerado que las escalas para ambos criterios no son las mismas), y siguiendo las etapas antes expuestas, se obtiene:

1. Se construye la matriz de decisión (tabla 1.3).
2. Se asignan pesos a los criterios (ver tabla 1.3).
3. La normalización $v_{ji} = \frac{x_{ji}}{\max x_{ji}}$ de los datos de partida, conduce a los resultados de la tabla 1.4.
4. Se determinan las acciones ideales y anti-ideales para cada criterio, y se obtienen los valores dados en la tabla 1.4.
5. El resultado del cálculo de las medidas de distancias Euclídeas d_j^+ y d_j^- , para cada acción respecto a las acciones ideal y anti-ideal, se ilustran en la tabla 1.5.
6. Los ratios de similitud D_j^+ obtenidos, se muestran en la tabla 1.5.
7. Por último, se ordenan las acciones en términos de la similitud D_j^+ . La ordenación O-TOPSIS obtenida se muestra en la tabla 1.5, y se compara con la ordenación O-Pesos, que se obtuvo con el mismo ejemplo aplicando el método de la suma ponderada.

Discusión

Si se utilizan valores diferentes de la métrica p , en el cálculo de las distancias L_p , los resultados (ordenaciones) finales pueden ser distintos. Yoon [213], mide la credibilidad de la función de distancias L_p , y concluye que la función distancias es menos creíble

Tabla 1.5: Distancias Euclídeas, similaridad y ordenaciones según TOPSIS y suma ponderada.

Acción	d_j^+	d_j^-	D_j^+	O-TOPSIS	O-Pesos
a	0.4286	0.1026	0.4631	2	2
b	0.2381	0.2564	0.4083	3	3
c	0.1190	0.4615	0.5369	1	1
d	0.2143	0.1795	0.2576	4	4

cuando el valor p aumenta, recomendando utilizar la métrica $p=1$ para obtener, desde un punto de vista matemático, la más creíble acción compromiso. De otra parte, Yoon apunta que cuando se normalizan las acciones, el valor de la métrica p afecta al grado de mutua interdependencia entre las acciones de los diferentes criterios, de manera que cuando p decrece, la interdependencia mutua entre las acciones se incrementa.

1.10.2 Métodos de agregación parcial

Los métodos de agregación parcial (según Vincke [203] *métodos de superación*, según Roy [163] y Maystre [131] *enfoque de superación de síntesis aceptando la incomparabilidad*), contrariamente a los métodos de agregación completa presentados en el apartado anterior, sí toleran la presencia de incomparabilidad e intransitividad.

Todos los métodos de agregación parcial, se inspiran en el concepto de *superación* definido por Roy [160] como *una relación binaria S definida en A tal que aSb si, dado aquello que se sabe de las preferencias del decisor y dada la calidad de las evaluaciones de las acciones y la naturaleza del problema, hay suficientes argumentos para admitir que a es al menos tan buena como b , sin que haya una razón importante para rechazar esta afirmación*. Para expresar las ideas de *al menos tan buena como* y *sin que haya una razón importante para rechazar esta afirmación* se tienen en cuenta los conceptos de *Concordancia* y *Discordancia*. La *Concordancia* cuantifica cuanto para un elevado número de criterios, una acción es preferida a otra, y para ello se fija un *Umbral Mínimo de Concordancia*. La *Discordancia* cuantifica hasta que punto no existe ningún criterio para el que una acción es mejor que otra, fijándose también un *Umbral Mínimo de Discordancia*.

Los trabajos pioneros en este dominio los constituyen los métodos ELECTRE⁹ (ELECTRE I [159], ELECTRE II [164], ELECTRE III [162], ELECTRE IV [166], ELECTRE IS [167] y ELECTRE TRI [215]). Todos ellos hacen uso directamente de la relación J de incomparabilidad y el sistema relacional de preferencias es de la

⁹Elimination Et Choix Traduisant la Réalité.

forma (S,J), donde la relación de superación S, no es antisimétrica ni necesariamente transitiva y la indiferencia se modela según: aSb y bSa .

También encontramos en la familia de los métodos de superación, a los métodos PROMETHEE¹⁰ [17][19][20][21]. Como sucedía con los métodos ELECTRE, aquí también se tolera la incomparabilidad J, aunque ahora no se hace uso de ella directamente, sino con ayuda de la relación (\approx) de no preferencia (se utiliza para modelar la indiferencia y la incomparabilidad sin diferenciarlas). El sistema relacional de preferencias es de la forma (P, \approx), donde la relación de preferencia P es antisimétrica y no necesariamente transitiva.

Existen también otros importantes trabajos que utilizan el enfoque de superación, ORESTE¹¹ [157], TACTIC¹² [198], etc. El método ELECTRE I es a continuación presentado.

Electre I

Es el primer método de superación que se publicó. ELECTRE I, se apoya en el concepto de verdadero criterio, bajo el cual subyace como se vio en la sección 1.8 una estructura de preferencia de preorden total. El método consiste en reducir el tamaño del conjunto de acciones potenciales. Para ello, divide el conjunto A de las acciones potenciales, en un subconjunto (núcleo N) de acciones incomparables entre sí (subconjunto más favorable para el centro decisor) y en otro subconjunto, que contiene acciones que son superadas por al menos una acción del núcleo N. La relación de superación se construye a partir de la noción de índices de concordancia y discordancia. Por último, un grafo visualiza la relación de superación para el conjunto de pares de acciones (a_j, a_n) del núcleo.

Presentación del método

Se tiene un conjunto de acciones $A = (a_1, a_2, \dots, a_j, \dots, a_n)$, un conjunto de criterios $G = (g_1, g_2, \dots, g_i, \dots, g_k)$ con una afectación de pesos w_i acorde a las preferencias del decisor, y una evaluación $x_{ji} = g_i(a_j)$ de la acción a_j bajo el criterio g_i (tabla 1.2). A partir de estos datos iniciales, se siguen los pasos siguientes:

1. Si los criterios tienen escalas diferentes, se normalizan los valores x_{ji} para obtener los valores v_{ji} .
2. Se establecen relaciones entre acciones efectuando comparaciones entre parejas de acciones (a_j, a_n) . De esta forma se obtienen los siguientes conjuntos de comparaciones:

¹⁰Preference Ranking Organisation Method for Enrichment Evaluations.

¹¹Organisation Rangement Et Synthèse des données relaTionElles

¹²Traitement des Alternatives Compte Tenu de l'Importance des Critères

- (a) $J^+(a_j, a_n) = \{i \in G / g_i(a_j) > g_i(a_n)\}$: conjunto de criterios para los cuales la acción a_j es preferida a la acción a_n .
- (b) $J^-(a_j, a_n) = \{i \in G / g_i(a_j) < g_i(a_n)\}$: conjunto de criterios para los cuales la acción a_n es preferida a la acción a_j .
- (c) $J^=(a_j, a_n) = \{i \in G / g_i(a_j) = g_i(a_n)\}$: conjunto de criterios para los cuales la acción a_j es indiferente a la acción a_n .
3. Se convierten los diferentes conjuntos obtenidos en el paso anterior, en valores numéricos. Así, se determina para cada conjunto, la suma de los pesos de los criterios que pertenecen a cada pareja de acciones (a_j, a_n) comparadas. De forma que:

(a) $W^+(a_j, a_n) = \sum_i w_i$ con $i \in J^+(a_j, a_n)$: suma de pesos de los criterios que

pertenecen al conjunto $J^+(a_j, a_n)$.

(b) $W^-(a_j, a_n) = \sum_i w_i$ con $i \in J^-(a_j, a_n)$: suma de pesos de los criterios que

pertenecen al conjunto $J^-(a_j, a_n)$.

(c) $W^=(a_j, a_n) = \sum_i w_i$ con $i \in J^=(a_j, a_n)$: suma de pesos de los criterios que

pertenecen al conjunto $J^=(a_j, a_n)$.

(d) $W(a_j, a_n) = W^+(a_j, a_n) + W^-(a_j, a_n) + W^=(a_j, a_n)$

4. Se fusionan los conjuntos obtenidos en los pasos anteriores, para obtener:

(a) $J(a_j, a_n) = J^+(a_j, a_n) \cup J^=(a_j, a_n)$: conjunto de concordancia.

(b) $J^-(a_j, a_n)$: conjunto de discordancia.

(c) C_{jn} : índice de concordancia que se calcula según la ecuación:

$$C_{jn} = \frac{W^+(a_j, a_n) + W^=(a_j, a_n)}{W(a_j, a_n)} \quad (1.10.5)$$

este índice, es un valor comprendido entre 0 y 1, y expresa en que medida la hipótesis $\ll a_j S a_n \gg$ (la acción a_j supera a la acción a_n), concuerda con la

realidad representada por la evaluación de las acciones. Una vez calculado el valor C_{jn} , es necesario tener un umbral de concordancia denotado por c y fijado por el decisor, que exprese el valor mínimo de concordancia requerido para que la hipótesis de partida sea considerada válida. De esta manera, si $C_{jn} \geq c$ significa que la importancia de los criterios para los cuales la acción $a_j Sa_n$, es suficientemente fuerte. Si $C_{jn} < c$ se rechaza la hipótesis de superación. Un valor típico para el umbral de concordancia es 0.7.

(d) D_{jn} : índice de discordancia que se calcula según:

$$D_{jn} \begin{cases} 0 & \text{si } J^-(a_j, a_n) = \phi \\ \frac{1}{\delta_i} \max (g_i(a_n) - g_i(a_j)) & \text{con } i \in J^-(a_j, a_n) \end{cases} \quad (1.10.6)$$

donde δ_i es la amplitud de la escala asociada al criterio i . Este índice varía entre 0 y 1, y expresa la medida de la oposición manifestada por los criterios discordantes a la aceptación de la hipótesis de partida. Calculado D_{jn} , se fija por el decisor el valor del umbral de discordancia, denotado por d , y que expresa el valor máximo de la discordancia requerida para que la hipótesis de partida $\ll a_j Sa_n \gg$ no sea rechazada. Si $D_{jn} \leq d$ la hipótesis de partida se aceptará y si $D_{jn} > d$ se rechazará. Un valor típico del valor de discordancia es 0.3.

5. Por último, se define la relación de superación entre las acciones: la acción a_j supera a la acción a_n si, de una parte, los criterios para los cuales la acción a_j es al menos tan buena como la acción a_n son suficientemente importantes y si, de otra parte la consideración del resto de los criterios no representa una oposición demasiado fuerte a dicha proposición. Esto se expresa por las siguientes condiciones:

$$a_j Sa_n \Leftrightarrow \begin{cases} C_{jn} \geq c \\ D_{jn} \leq d \end{cases}$$

La relación de superación se puede expresar gráficamente mediante un grafo de superación, a partir del cual se llega al subconjunto N de acciones incomparables.

Ejemplo

Tomando de nuevo el problema de Vallin presentado en la sección 1.10.1, y siguiendo los pasos que se acaban de describir tenemos:

Tabla 1.6: Comparaciones entre acciones para el criterio g_1 (izquierda) y comparaciones entre acciones para el criterio g_2 (derecha).

	a	b	c	d
a	0	J^+	J^+	J^+
b	J^-	0	J^+	J^+
c	J^-	J^-	0	J^-
d	J^-	J^-	J^+	0

	a	b	c	d
a	0	J^-	J^-	J^-
b	J^+	0	J^-	J^+
c	J^+	J^+	0	J^+
d	J^+	J^-	J^-	0

Tabla 1.7: Resumen de los conjuntos J^+ (izquierda), $J^=$ (centro) y J^- (derecha).

	a	b	c	d
a	0	{1}	{1}	{1}
b	{2}	0	{1}	{1,2}
c	{2}	{2}	0	{2}
d	{2}	{0}	{1}	0

	a	b	c	d
a	0	{0}	{0}	{0}
b	{0}	0	{0}	{0}
c	{0}	{0}	0	{0}
d	{0}	{0}	{0}	0

	a	b	c	d
a	0	{2}	{2}	{2}
b	{0}	0	{2}	{0}
c	{0}	{0}	0	{0}
d	{0}	{1,2}	{2}	0

1. Se elabora la matriz de decisión (tabla 1.3).
2. Se establecen los pesos a los criterios (ver tabla 1.3).
3. Se efectúan comparaciones entre parejas de acciones (a_j, a_n) por criterio (tablas 1.6 izquierda y derecha) y a continuación se obtienen los conjuntos J^+ , $J^=$ y J^- que se muestran en las tablas 1.7 izquierda, centro y derecha respectivamente.
4. Los conjuntos J^+ , $J^=$ y J^- obtenidos en el paso anterior, se convierten en valores numéricos para obtener los conjuntos W^+ , $W^=$ y W^- dados en las tablas 1.8 izquierda, centro y derecha respectivamente.

Tabla 1.8: Resumen de coeficientes W^+ (izquierda), $W^=$ (centro) y W^- (derecha).

	a	b	c	d
a	0	0.5	0.5	0.5
b	0.5	0	0.5	1
c	0.5	0.5	0	0.5
d	0.5	0	0.5	0

	a	b	c	d
a	0	0	0	0
b	0	0	0	0
c	0	0	0	0
d	0	0	0	0

	a	b	c	d
a	0	0.5	0.5	0.5
b	0.5	0	0.5	0
c	0.5	0.5	0	0.5
d	0.5	1	0.5	0

Tabla 1.9: Coeficientes de concordancia C_{jn} (zquierda) y discordancia D_{jn} (derecha).

	a	b	c	d
a	0	0.5	0.5	0.5
b	0.5	0	0.5	1
c	0.5	0.5	0	0.5
d	0.5	0	0.5	0

	a	b	c	d
a	0	0.3	0.7	0.15
b	0.4	0	0.4	0
c	0.65	0.25	0	0.2
d	0.45	0.05	0.55	0



Figura 1.13: Grafo de relaciones de superación entre las acciones b y d.

5. A partir de los conjuntos obtenidos en los pasos anteriores, se pueden calcular:
 - (a) El conjunto de concordancia $J(a_j, a_n) = J^+(a_j, a_n) \cup J^-(a_j, a_n)$, que en este ejemplo coincide con los valores de la tabla 1.7 izquierda.
 - (b) El conjunto de discordancia tabla 1.7 derecha.
 - (c) Los coeficientes de concordancia C_{jn} a partir de la ecuación (1.10.5) y que se muestran en la tabla 1.9 izquierda.
 - (d) Los coeficientes de discordancia D_{jn} a partir de la ecuación (1.10.6) y que se muestran en la tabla 1.9 derecha.

6. Se fijan los umbrales de concordancia y discordancia a los valores $c=0.7$ y $d=0.3$, y a partir de las tablas 1.9 izquierda y 1.9 derecha, se puede decir que la acción b *supera* a la acción d (bSd) pues satisface los valores $C_{bd} \geq 0.7$ y $D_{bd} \leq 0.3$. Por tanto bSd es la única relación que se obtiene (ver grafo de la figura 1.13). Como conclusión, se puede afirmar que la acción b es la mejor de todas las acciones.

Discusión

Este método, formaliza a partir de los criterios definidos, el razonamiento humano que se habría hecho de forma natural. El método se adapta a la problemática P_α o problemática de elección, pudiéndose aplicar a un considerable número de acciones y obtener relaciones entre ellas, que hechas a mano resultaría muy laborioso. De otro parte, el número de etapas que deben realizarse para llegar al resultado final de

elección es importante, además el decisor debe definir los valores para los umbrales de concordancia y discordancia, y si estos se cambian, es de esperar que se obtengan resultados diferentes.

Capítulo 2

Optimización Multiobjetivo

2.1 Multiobjetivo-Multicriterio: breve discusión

Como se apuntó en la sección 1.1, por los problemas *Multicriterio* se interesa la AMD, la cual tiene a sus principales representantes en las escuelas de economía y dirección de empresas. Aquí, el interés es la *toma de decisiones* y el problema tipo consiste en *ayudar a elegir a un decisor, según varios criterios, entre varias acciones*.

Progresivamente, muchas de las técnicas de la AMD, se han ido extendiendo a las ciencias y las ingenierías, donde más que la *decisión*, lo que se investiga es la *optimización*. Aquí lo que se pretende, es *buscar con rápidos algoritmos, un conjunto¹ de soluciones eficientes*. Ahora se prefiere hablar de *Optimización Multiobjetivo*.

Por tanto, cuando hablamos de *Ayuda Multicriterio a la Decisión* y de *Optimización Multiobjetivo*, estamos hablando de dos comunidades diferentes, que se comunican entre si, y que tienen competencias distintas. No obstante, si se examina la literatura, se observa que no existe una gran diferencia de fondo entre los métodos de decisión y los métodos de optimización, y los términos *Multicriterio* y *Multiobjetivo* se utilizan indistintamente, en especial en la comunidad de las ciencias y las ingenierías.

2.2 Clasificación de los métodos de Optimización Multiobjetivo

Cuando nos enfrentamos a *Problemas de Optimización Multiobjetivo* (POM), nos encontramos que la mayoría de ellos son calificados como difíciles². Numerosos métodos

¹*Este conjunto es también denominado frente de Pareto.*

²En optimización [36][59], se dice que un problema con variables continuas es difícil, cuando no se conoce algoritmo capaz de identificar un óptimo global en un número finito de cálculos. Si se trata de un problema con variables discretas se dice que es NP-difícil, cuando la resolución exacta no es posible en un tiempo proporcional a N^n , siendo N el número de parámetros desconocidos del

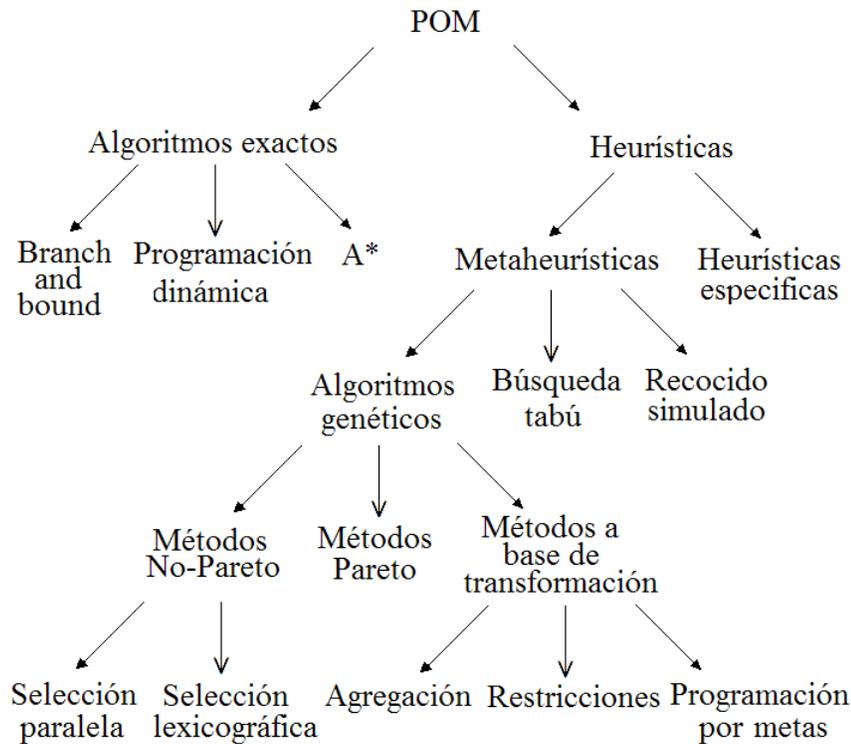


Figura 2.1: Clasificación de los métodos de Optimización Multiobjetivo.

han sido desarrollados con la intención de dar respuesta a este tipo de problemas.

Basseur et al. [11] y Munier [141], clasifican los métodos de Optimización Multiobjetivo (figura 2.1) en *métodos exactos* tales como el Branch and Bound [180] [204], el Algoritmo A* [186] y la Programación Dinámica [24], y *métodos heurísticos*. Los métodos heurísticos son a su vez divididos en dos clases: *heurísticas específicas* [81], que utilizan conocimientos del dominio del problema que se trata, y *metaheurísticas* o algoritmos más generales para la resolución de una gran variedad de POM, entre ellas la *Búsqueda Tabú* [83][84], el *Recocido Simulado* [111] y los *Algoritmos Genéticos* [85][98]. Finalmente los Algoritmos Genéticos son divididos en métodos a base de transformación, métodos no-Pareto y métodos Pareto.

Según Van Veldhuizen y Lamont [200], la solución final de un POM es resultado de dos procesos: la *optimización* y la *decisión*, y dependiendo de si las preferencias del decisor son expresadas antes, durante o después del proceso de optimización, dividen los métodos de Optimización Multiobjetivo (ver también [11][36][130][142]) en tres familias:

problema, y n una constante entera.

- Métodos a priori. El decisor combina los diferentes objetivos en una función escalar.
- Métodos progresivos. La toma de decisión y la optimización son interactivas.
- Métodos a posteriori. El decisor elige a partir de un conjunto de soluciones eficientes.

Otra clasificación de los métodos de Optimización Multiobjetivo la hacen Collette y Siarry [36], dividiendo los métodos en cinco grupos:

- Métodos escalares (o agregativos). Todos estos métodos transforman el problema Multiobjetivo inicial, en un problema de optimización mono-objetivo.
- Métodos interactivos. Un método interactivo es un proceso secuencial compuesto de varias iteraciones, de forma que un decisor puede articular sus preferencias en cada etapa del proceso.
- Métodos difusos. Estos métodos hacen intervenir la lógica difusa desarrollada por Zadeh [217], la cual permite tratar la incertidumbre y la imprecisión así como las transiciones progresivas entre lo *verdadero* y lo *falso*.
- Métodos que utilizan una metaheurística. Son métodos estocásticos, diseñados para resolver problemas de optimización difícil, mediante una aproximación intuitiva.
- Métodos de ayuda a la decisión. Estos métodos se basan en el establecimiento de una relación de orden entre las diferentes acciones o soluciones.

2.3 Conceptos básicos

2.3.1 Problema Multiobjetivo

Un POM es aquel que corresponde a una cierta realidad industrial, económica u otra y sobre el que un decisor desea optimizar simultáneamente varios objetivos usualmente en conflicto entre si. De manera general un POM se puede representar de la siguiente forma:

en términos de minimización (en lo que sigue, se considera que todas las funciones son a minimizar)

$$\left\{ \begin{array}{ll} \text{mín.} & F(x) = f_1(x), f_2(x), \dots, f_k(x) \\ \text{s.a.} & \\ & g_j(x) \leq 0 \quad j \in (1, 2, \dots, m) \\ & h_q(x) = 0 \quad q \in (1, 2, \dots, p) \\ & B_{\text{inf}} \leq x_l \leq B_{\text{sup}} \quad l \in (1, 2, \dots, n) \end{array} \right. \quad (2.3.1)$$

$F(x)$ es el vector de objetivos a minimizar y $k \geq 2$ representa el número de objetivos.

Los valores B_{inf} y B_{sup} son respectivamente los límites inferior y superior de la variable x_l . Los valores de x_l que verifican estas restricciones definen el *espacio de búsqueda* (figura 2.2 izquierda).

Las restricciones $g_j(x) \leq 0$ y $h_j(x) = 0$ representan respectivamente m restricciones de desigualdad y p restricciones de igualdad. Los valores de x que satisfacen el conjunto de las $(m+p)$ restricciones, definen el *espacio realizable* S (figura 2.2 centro).

El vector $x = (x_1, x_2, \dots, x_n) \in S$ es un vector solución de n variables de decisión.

En el marco de la Optimización Multiobjetivo, con frecuencia el decisor razona en términos de la evaluación de una solución según cada objetivo, situándose en el espacio de los objetivos. Para representar el conjunto de soluciones realizables, en el espacio de los objetivos, es preciso determinar la imagen de cada solución realizable del espacio de decisión. Obtenemos así un conjunto $Z = f(S)$ (figura 2.2 derecha) definido como:

$$Z = \{z = (z_1, z_2, \dots, z_k) \in R^k : z_1 = f_1(x), z_2 = f_2(x), \dots, z_k = f_k(x), \forall x \in S\}$$

donde Z representa el conjunto de soluciones realizables en el espacio de los objetivos y $z = (z_1, z_2, \dots, z_k)$ una solución realizable en el espacio de los objetivos.

Siempre que se haga mención al espacio de decisión o al espacio de los objetivos, se supondrá la porción realizable.

2.3.2 Dominancia

Cuando se resuelve el POM (2.3.1), se obtiene una gran cantidad de soluciones y solo un número restringido de ellas interesa. Para que una solución resulte de interés, es preciso que exista una *relación de dominancia* entre la solución considerada y el resto de soluciones.

Definición 1. (Dominancia de Pareto) Una solución $x^t = (x_1^t, x_2^t, \dots, x_n^t)$ domina una solución $x^u = (x_1^u, x_2^u, \dots, x_n^u)$ si se verifican las siguientes condiciones:

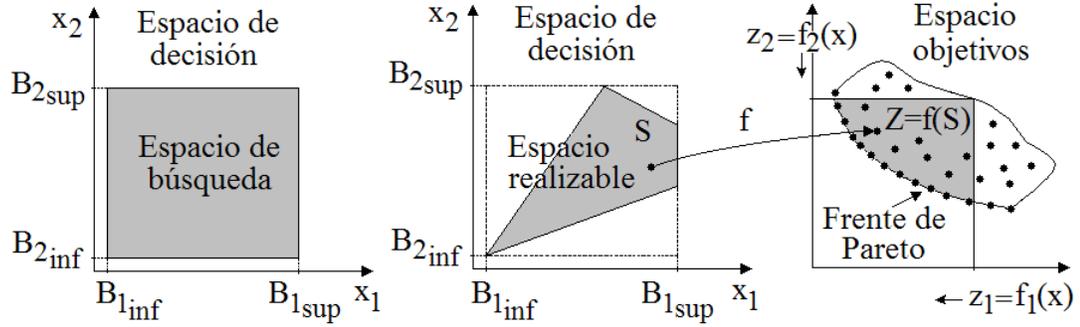


Figura 2.2: Espacio de búsqueda (izquierda). Espacio realizable (centro). Espacio de los objetivos (derecha).

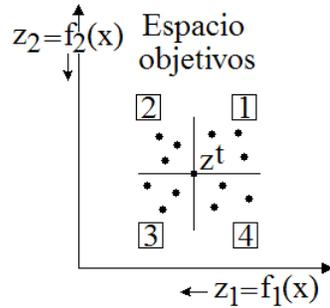


Figura 2.3: Relaciones posibles de dominancia entre soluciones.

1. $z_i^t \leq z_i^u \quad \forall i \in (1, 2, \dots, k)$
2. $\exists i \in (1, 2, \dots, k)$ tal que $z_i^t < z_i^u$

Cuando a una solución z^t se le aplica el concepto de dominancia de Pareto, se pueden definir cuatro regiones. Por ejemplo, en la figura 2.3, tomando la solución z^t definida en el espacio de los objetivos como referencia:

- z^t domina las soluciones contenidas en la región 1.
- z^t es incomparable con las soluciones contenidas en la región 2.
- z^t es dominada por las soluciones contenidas en la región 3.
- z^t es incomparable con las soluciones contenidas en la región 4.

Como se vio en la sección 1.7.3, la relación de dominancia de Pareto es una relación de orden parcial (I,P,J).

Definición 2. (Pareto óptima) Una solución $x^* \in S$ es Pareto óptima³ si y solo si no existe una solución $x \in S$ tal que $x \in S$ domine $x^* \in S$. Una solución Pareto óptima también es conocida como solución eficaz o solución no-dominada.

Definición 3. (Conjunto de Pareto) Habitualmente, en un POM no existe una única solución $x^* \in S$ Pareto óptima, si más bien un conjunto de soluciones denominado conjunto de Pareto o conjunto eficiente.

Definición 4. (Frente de Pareto) El conjunto de soluciones en el espacio de los objetivos que son imágenes del conjunto de Pareto, se denomina frente de Pareto (figura 2.2 derecha).

Definición 5. (Soluciones equivalentes) Dos soluciones $x^t = (x_1^t, x_2^t, \dots, x_n^t)$ y $x^u = (x_1^u, x_2^u, \dots, x_n^u)$ con $x^t \neq x^u$, se dicen equivalentes si se verifica:

$$1. z_i^t = z_i^u \quad \forall i \in (1, 2, \dots, k)$$

2.3.3 Relaciones derivadas de la dominancia.

Dominancia débil de Pareto

Una solución $x^t = (x_1^t, x_2^t, \dots, x_n^t)$ domina débilmente una solución $x^u = (x_1^u, x_2^u, \dots, x_n^u)$ si se verifica la siguiente condición:

$$1. z_i^t \leq z_i^u \quad \forall i \in (1, 2, \dots, k)$$

Como se vio en la sección 1.7.3, la relación de dominancia débil de Pareto es una relación de preorden parcial (I,P,J).

ϵ -Dominancia

La ϵ -dominancia [123], es una forma relajada de la dominancia de Pareto y se define de la siguiente forma. En el espacio de los objetivos del problema (2.3.1), una solución $z^t = (z_1^t, z_2^t, \dots, z_k^t)$ ϵ -domina la solución $z^u = (z_1^u, z_2^u, \dots, z_k^u)$ para algún $\epsilon > 0$, si se verifica que:

$$1. \exists i \in (1, 2, \dots, k) \text{ tal que } (1 - \epsilon)z_i^t \leq z_i^u$$

El concepto de ϵ -dominancia, se ilustra en la figura 2.4 izquierda, en comparación con el concepto de dominancia de Pareto figura 2.4 derecha.

³El concepto de óptimo Pareto lo formuló Wilfredo Pareto [149] a finales del siglo XIX.

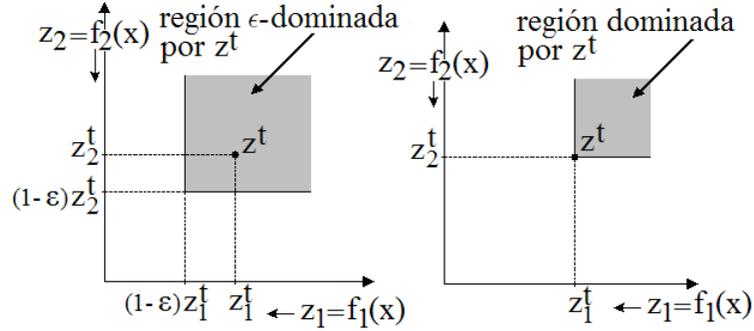


Figura 2.4: Concepto de ϵ -dominancia (izquierda). Dominancia de Pareto (derecha).

Otras relaciones de dominancia

Otras relaciones de dominancia se pueden encontrar en la literatura, son ejemplos entre otras: la dominancia lexicográfica [60], la dominancia de Geoffrion [61], la cono-dominancia [36], la α -dominancia [148].

2.3.4 Puntos de referencia

Asociados al frente de Pareto, se distinguen varios puntos de referencia (vectores de referencia o soluciones de referencia) que se definen a continuación:

Definición 6. Punto Ideal⁴ (z^I), es un vector solución $z^I = (z_1^I, z_2^I, \dots, z_k^I)$, construido con los mejores valores función objetivo de todas las soluciones del espacio de los objetivos (figura 2.5 izquierda). Este punto es en general inalcanzable.

Definición 7. Punto Nadir (z^N), es un vector solución $z^N = (z_1^N, z_2^N, \dots, z_k^N)$, construido con los peores valores función objetivo de todas las soluciones del frente de Pareto (figura 2.5 izquierda). Este punto puede ser alcanzable o inalcanzable.

Definición 8. Punto de Referencia (z^*), es un vector solución $z^* = (z_1^*, z_2^*, \dots, z_k^*)$, que define la meta a alcanzar por cada función objetivo.

Definición 9. Punto de Funcionamiento (PF) [134][136], se define como aquel vector solución $x^{PF} = (x_1^{PF}, x_2^{PF}, \dots, x_n^{PF})$ de un problema real Multiobjetivo, que identifica una solución estable de funcionamiento (o trabajo), no necesariamente efectiva o Pareto óptima. El PF también se define como aquel vector solución $x^{PF} =$

⁴También: Punto utópico [130], Punto de referencia [210], Vector ideal [158][11], Punto diana [95], Punto de mira [160].

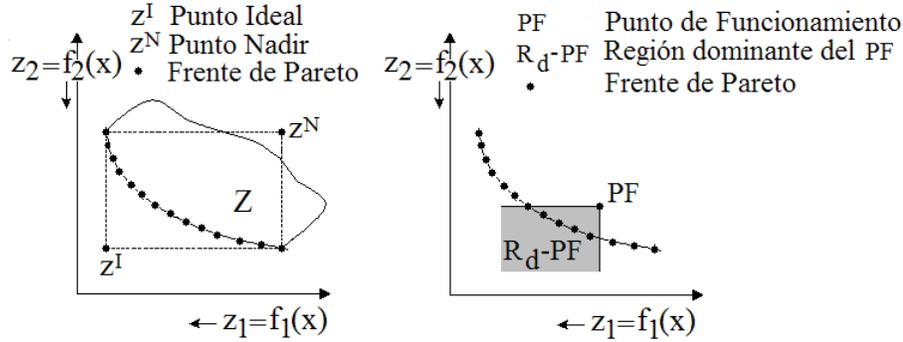


Figura 2.5: Puntos Ideal y Nadir (izquierda). Punto de Funcionamiento (PF) y Región dominante del Punto de Funcionamiento (R_d -PF) (derecha).

$(x_1^{PF}, x_2^{PF}, \dots, x_n^{PF})$ de un problema real Multiobjetivo, que determina el estado actual del sistema. La imagen de x^{PF} , define el PF en el espacio de los objetivos ($z^{PF} = (z_1^{PF}, z_2^{PF}, \dots, z_n^{PF})$), ver la figura 2.5 derecha.

Se define la Región dominante del PF (R_d -PF) [134], como aquel subconjunto de soluciones (subregión) del frente de Pareto, que dominan al PF (figura 2.5 derecha).

Axioma 1. *Axioma de pertenencia.* Si una solución $x^{PF} = (x_1^{PF}, x_2^{PF}, \dots, x_n^{PF})$ definida en el espacio de decisión, es el PF de un problema real Multiobjetivo, entonces $x^{PF} = (x_1^{PF}, x_2^{PF}, \dots, x_n^{PF})$ pertenece al conjunto realizable de soluciones en el espacio de decisión S .

Axioma 2. *Axioma de pertenencia [134].* Si una solución $z^{PF} = (z_1^{PF}, z_2^{PF}, \dots, z_k^{PF})$ definida en el espacio de los objetivos, es el PF de un problema real Multiobjetivo, entonces $z^{PF} = (z_1^{PF}, z_2^{PF}, \dots, z_k^{PF})$ pertenece al conjunto realizable de soluciones en el espacio de los objetivos Z .

2.4 Métodos que articulan preferencias a posteriori

Muchos de estos métodos se caracterizan porque transforman el POM (2.3.1) en un problema uniobjetivo. La mayoría de las veces, el método consiste en hacer variar arbitrariamente los parámetros de la función utilizada (por ejemplo los pesos), con la finalidad de generar un conjunto de soluciones que aproxime el frente de Pareto. Debe observarse que los parámetros elegidos no guardan necesariamente relación alguna con las preferencias del decisor. Métodos representativos de estas técnicas son

el método de los pesos [36][142][216], el método de la distancia a un punto de referencia [36][142], el método de las restricciones [36][129][142], el método de Corley [37], physical programming [139][140],... Se detallan a continuación los tres primeros métodos.

2.4.1 Métodos de los pesos

El método de los pesos, consiste en transformar el POM (2.3.1) en un problema con un solo objetivo, para ello se agregan los distintos objetivos en forma de una suma ponderada:

$$\left\{ \begin{array}{l} \text{mín. } F_w(x) = \sum_{i=1}^k w_i f_i(x) = w_1 f_1(x) + w_2 f_2(x) + \dots + w_k f_k(x) \\ \text{s.a.} \\ g_j(x) \leq 0 \quad j \in (1, 2, \dots, l) \\ h_q(x) = 0 \quad q \in (1, 2, \dots, m) \\ B_{\text{inf}} \leq x_l \leq B_{\text{sup}} \quad l \in (1, 2, \dots, n) \end{array} \right. \quad (2.4.1)$$

donde los pesos $w_i \geq 0$ son tales que $\sum_{i=1}^k w_i = 1$.

En [216], se demuestra que si todos los pesos w_i son positivos, entonces para un vector de pesos fijo, la solución del problema (2.4.1) es una solución Pareto óptima del problema (2.3.1). Si se quiere obtener un conjunto que contenga un gran número de soluciones Pareto Óptimas, es necesario resolver varias veces el problema (2.4.1) cambiando cada vez el valor del vector pesos.

El funcionamiento del método se puede representar gráficamente a partir del siguiente problema:

$$\left\{ \begin{array}{l} \text{mín. } f_1(x) \\ \text{mín. } f_2(x) \\ \text{s.a.} \\ g(x) \leq 0 \\ h(x) = 0 \end{array} \right. \quad (2.4.2)$$

agregando los dos objetivos en forma de suma ponderada, obtenemos una nueva expresión:

$$F_w(x) = w_1 f_1(x) + w_2 f_2(x) \quad (2.4.3)$$

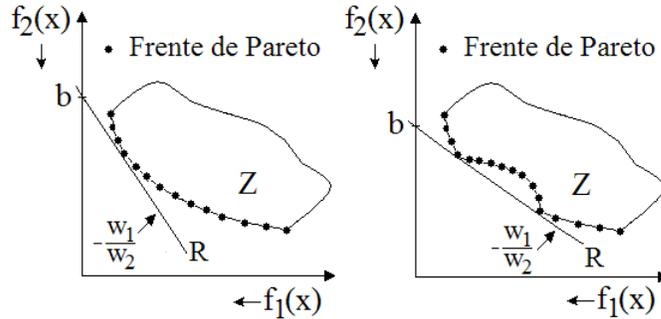


Figura 2.6: Interpretación gráfica del método de los pesos (izquierda). Dificultad para el método de los pesos (derecha).

despejando $f_2(x)$ en la expresión (2.4.3) obtenemos:

$$f_2(x) = -\frac{w_1}{w_2}f_1(x) + \frac{F_w(x)}{w_2} \quad (2.4.4)$$

como lo que se busca es minimizar $F_w(x)$, en realidad se busca minimizar la constante b de la recta:

$$f_2(x) = -\frac{w_1}{w_2}f_1(x) + b \quad (2.4.5)$$

En la figura 2.6 (izquierda), se representa la recta R que corresponden a un vector de pesos (w_1, w_2) fijado. Z representa el conjunto de soluciones realizables en el espacio de los objetivos del problema (2.4.2).

El método consiste en encontrar el valor b mínimo de la recta dada por (2.4.5), de forma que esta sea tangente al conjunto Z .

Ejemplo

Sea el problema F_2 de Schaffer [174]:

$$\begin{cases} \text{mín. } f_1(x) = x^2 \\ \text{mín. } f_2(x) = (x - 2)^2 \\ \text{s.a.} \\ x \in [0, 2] \end{cases} \quad (2.4.6)$$

la representación de las funciones $f_1(x)$ y $f_2(x)$ se muestran en la figura 2.7.

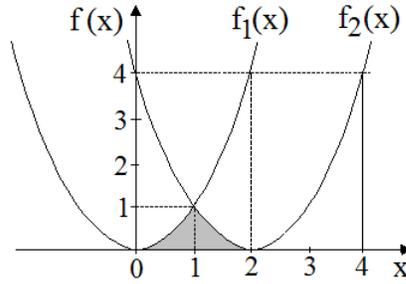


Figura 2.7: Problema F_2 de Schaffer.

Tabla 2.1: Resultados obtenidos al aplicar el método de los pesos (y variar el vector pesos w_i) al problema F_2 de Schaffer.

w_2	0	0.2928	0.5	0.7071	1
w_1	1	0.7071	0.5	0.2928	0
x^*	0	0.5857	1	1.4142	2
$f_1(x^*)$	0	0.3431	1	2	4
$f_2(x^*)$	4	2	1	0.3431	0

Se busca minimizar la siguiente función:

$$\left\{ \begin{array}{l} \text{mín. } F_w(x) = w_1x^2 + w_2(x - 2)^2 \\ \text{s.a.} \\ w_1, w_2 \geq 0 \\ w_1 + w_2 = 1 \\ x \in [0, 2] \end{array} \right. \quad (2.4.7)$$

Derivando e igualando a cero (2.4.7) $\frac{dF_w(x)}{dx}=0$, se obtiene que:

$$x^* = 2w_2 \quad (2.4.8)$$

además este valor de x^* es mínimo pues $\frac{d^2F_w(x)}{dx^2} = 2 > 0$.

Sustituyendo el juego de pesos w_i dados en la tabla 2.1 en (2.4.8), se obtienen los valores $f_1(x^*), f_2(x^*)$ (ver también tabla 2.1), que conforman un conjunto de soluciones óptimas (frente de Pareto) y que se muestran en la figura 2.8.

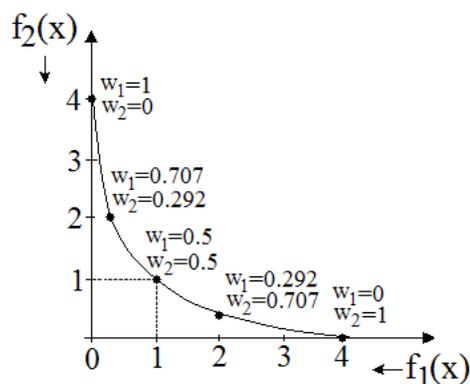


Figura 2.8: Ilustración gráfica en el espacio de los objetivos, de las soluciones obtenidas al aplicar el método de los pesos al problema F_2 de Schaffer.

Discusión

El método es sencillo de utilizar, aunque presenta dos inconvenientes importantes. El primero es que para problemas no lineales, el hecho de elegir vectores de pesos uniformemente distribuidos, no se garantiza obtener un conjunto de soluciones bien repartidas sobre el frente de Pareto. El segundo inconveniente proviene del hecho que para problemas cuyo frente de Pareto contiene regiones no convexas, el método no permite encontrar soluciones situadas en tales regiones. La figura 2.6 (derecha) muestra esta situación.

2.4.2 Método de la distancia a un punto de referencia

Con este método todas las funciones objetivo del POM (2.3.1), son combinadas para formar una sola función objetivo, la cual puede verse como una función de distancias que intenta minimizar la distancia entre los puntos solución y un punto de referencia en el espacio de los objetivos. Una medida de distancia (ver [36][137][214][218], ver también el método TOPSIS en la sección 1.10.1) comúnmente utilizada en estos métodos es la siguiente:

$$L_p(x) = \left[\sum_{i=1}^k w_i |f_i(x) - F_i^*|^p \right]^{\frac{1}{p}} \quad (2.4.9)$$

Así el POM (2.3.1) es transformado en el problema uniobjetivo siguiente:

$$\left\{ \begin{array}{l} \text{mín. } L_p(x) = [w_1 |f_1(x) - F_1^*|^p + w_2 |f_2(x) - F_2^*|^p + \dots, w_k |f_k(x) - F_k^*|^p]^{\frac{1}{p}} \\ \text{s.a.} \\ g_j(x) \leq 0 \quad j \in (1, 2, \dots, l) \\ h_q(x) = 0 \quad q \in (1, 2, \dots, m) \\ B_{\text{inf}} \leq x_l \leq B_{\text{sup}} \quad l \in (1, 2, \dots, n) \end{array} \right. \quad (2.4.10)$$

$L_p(x)$ es la familia de métricas de la función distancia, la cual se calcula para un valor particular de p ($1 \leq p \leq \infty$), valores comunes para p son: $p=1$ (distancia Manhattan), $p=2$ (distancia Euclidiana) y $p=\infty$ (distancia Tchebycheff). F_i^* con $i \in (1, 2, \dots, k)$,

es un punto o vector de referencia. El vector de pesos w_i ($w_i \geq 0$ y $\sum_{i=1}^k w_i = 1$)

permite encontrar para cada resolución del problema (2.4.10), una solución en una cierta dirección.

Al igual que con el método de los pesos, se aplica el método de la distancia a un punto de referencia al problema (2.4.2), para mejor entender su funcionamiento. Si por ejemplo, consideramos la distancia euclidiana ($p=2$), la agregación de las funciones objetivo conduce a la expresión:

$$L_2(x) = [w_1 |f_1(x) - F_1^*|^2 + w_2 |f_2(x) - F_2^*|^2]^{\frac{1}{2}} \quad (2.4.11)$$

Se pretende minimizar $L_2(x)$, que equivale a minimizar la constante c , en la ecuación de la elipse de centro (F_1^*, F_2^*) :

$$\frac{(f_1(x) - F_1^*)^2}{\frac{c}{w_1}} + \frac{(f_2(x) - F_2^*)^2}{\frac{c}{w_2}} = 1 \quad (2.4.12)$$

El método trata de encontrar el valor mínimo c en la ecuación de la elipse (2.4.12), de forma que tangente con el dominio Z (figura 2.9).

Ejemplo

Retomando el problema F_2 de Schaffer (2.4.6), si se elige como punto de referencia $F_1^* = 0$, $F_2^* = 0$ y el valor de la métrica elegida es $p=2$, lo que se busca es minimizar

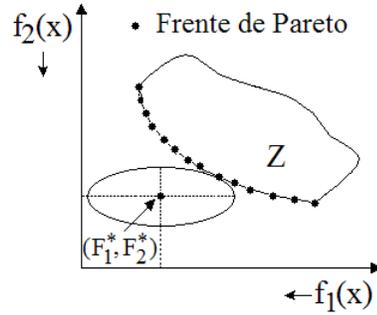


Figura 2.9: Interpretación gráfica del método de la distancia a un punto de referencia.

Tabla 2.2: Resultados obtenidos al aplicar el método de la distancia a un punto de referencia (y variar el vector pesos w_i) al problema F_2 de Schaffer.

w_2	0	0.0664	0.5	0.9336	1
w_1	1	0.9336	0.5	0.0664	0
x^*	0	0.5857	1	1.4142	2
$f_1(x^*)$	0	0.3431	1	2	4
$f_2(x^*)$	4	2	1	0.3431	0

la función:

$$\left\{ \begin{array}{l} \text{mín. } L_2(x) = w_1 x^4 + w_2 (x - 2)^4 \\ \text{s.a.} \\ w_1, w_2 \geq 0 \\ w_1 + w_2 = 1 \\ x \in [0, 2] \end{array} \right. \quad (2.4.13)$$

Derivando e igualando a cero (2.4.13) se obtiene que:

$$\frac{dL_2(x)}{dx} = 0 = 4w_1 x^3 + 4w_2 (x - 2)^3 \quad (2.4.14)$$

La ecuación (2.4.14) se satisface para los valores de los vectores pesos w_i y del vector solución x^* mostrados en la tabla 2.2. Los valores $f_1(x^*), f_2(x^*)$ configuran el frente de Pareto mostrado en la figura 2.10.

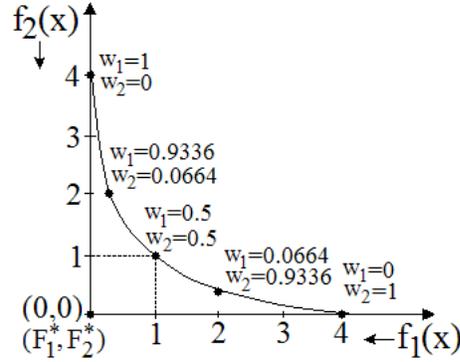


Figura 2.10: Ilustración gráfica en el espacio de los objetivos, de las soluciones obtenidas al aplicar el método de la distancia a un punto de referencia al problema F_2 de Schaffer.

Discusión

Con el aumento del valor de la métrica p , es posible encontrar soluciones que pertenezcan a regiones no convexas [158], es más, utilizando la métrica de Tchebycheff y diferentes valores de vectores de pesos, es teóricamente posible encontrar todas las soluciones del problema (2.3.1). Sin embargo, hay que remarcar que cuando el valor de p crece, la función objetivo se hace no diferenciable. Además para que este método funcione bien, se debe ser capaz de elegir un buen punto de referencia, si no, se corre el riesgo de no encontrar todas las soluciones del frente de Pareto del problema.

2.4.3 Método de las ϵ -restricciones

Otra forma de transformar el POM (2.3.1) en otro uniobjetivo, consiste en convertir $k-1$ objetivos (hay k objetivos) del problema en restricciones, y minimizar o maximizar el objetivo restante. De esta manera, el POM (2.3.1) se reformula de la siguiente forma:

$$\left\{ \begin{array}{l} \text{mín. } f_1(x) \\ \text{s.a.} \\ f_2(x) \leq \epsilon_2 \\ \cdot \\ \cdot \\ f_k(x) \leq \epsilon_k \\ g_j(x) \leq 0 \quad j \in (1, 2, \dots, l) \\ h_q(x) = 0 \quad q \in (1, 2, \dots, m) \\ B_{\text{inf}} \leq x_l \leq B_{\text{sup}} \quad l \in (1, 2, \dots, n) \end{array} \right. \quad (2.4.15)$$

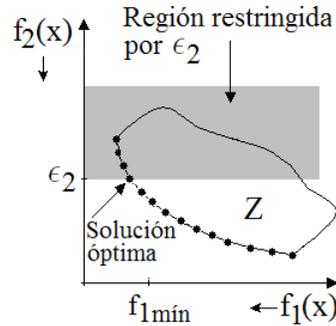


Figura 2.11: Ilustración gráfica del funcionamiento del método de las ϵ -restricciones.

Para encontrar el conjunto de soluciones Pareto óptimas, se resuelve varias veces el problema POM (2.4.15) eligiendo diferentes valores del vector restricciones ϵ_i con $i \in (2, 3, \dots, k)$.

El funcionamiento del método, se puede ver gráficamente reformulando el problema bi-objetivo (2.4.2) según:

$$\begin{cases} \text{mín. } f_1(x) \\ \text{s.a.} \\ f_2(x) \leq \epsilon_2 \\ g_j(x) \leq 0 & j \in (1, 2, \dots, l) \\ h_q(x) = 0 & q \in (1, 2, \dots, m) \end{cases} \quad (2.4.16)$$

y se ilustra en la figura 2.11 (la elección de un valor ϵ_2 hace inaccesible una determinada región del espacio de los objetivos y encuentra una solución realizable).

Ejemplo

Retomando una vez más el problema F_2 de Schaffer (2.4.6) y aplicando el método de las ϵ -restricciones, se obtiene el siguiente problema:

$$\begin{cases} \text{mín. } x^2 \\ \text{s.a.} \\ (x - 2)^2 \leq \epsilon_2 \\ x \in [0, 2] \end{cases} \quad (2.4.17)$$

Elijiendo los valores $\epsilon = 0, 1, 2, 3$ y 4 , obtenemos los resultados que se muestran en la tabla 2.3 y en la figura 2.12.

Tabla 2.3: Resultados obtenidos al aplicar el método de las ϵ -restricciones (y variar el vector ϵ) al problema F_2 de Schaffer.

ϵ	x^*	$f_1(x^*)$	$f_2(x^*)$
0	2	4	0
1	1	1	1
2	0.5858	0.3431	2
3	0.2679	0.0718	3
4	0	0	4

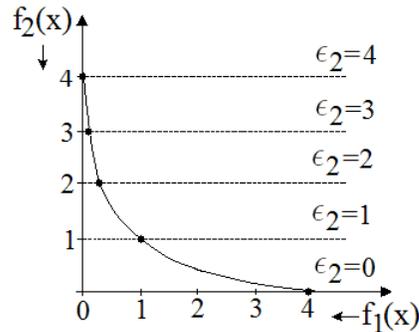


Figura 2.12: Ilustración gráfica en el espacio de los objetivos, de las soluciones obtenidas al aplicar el método de las ϵ -restricciones al problema F_2 de Schaffer.

Discusión

El método es de una relativa simplicidad, y tiene la capacidad de encontrar soluciones Pareto óptimas que pertenecen a regiones no convexas. Sin embargo, el método presenta los siguientes inconvenientes: la carga computacional es considerable, se puede elegir un valor ϵ para el cual no existe solución realizable (en la figura 2.13, valores $\epsilon_2 < \epsilon_2^t$ no obtienen una solución realizable) y por último, se pueden elegir valores superiores a un cierto ϵ y encontrar siempre la misma solución realizable (en la figura 2.13, valores $\epsilon_2 > \epsilon_2^u$ siempre conducen a la misma solución u).

2.5 Métodos que articulan preferencias a priori

En ocasiones el centro decisor puede tener un buen conocimiento del problema a tratar. La idea de los métodos a priori, es que ese conocimiento, especificado en forma de preferencias, se utilice en el proceso de búsqueda del conjunto de soluciones óptimas. Métodos que incorporan a priori preferencias ya sean expresándolas en forma

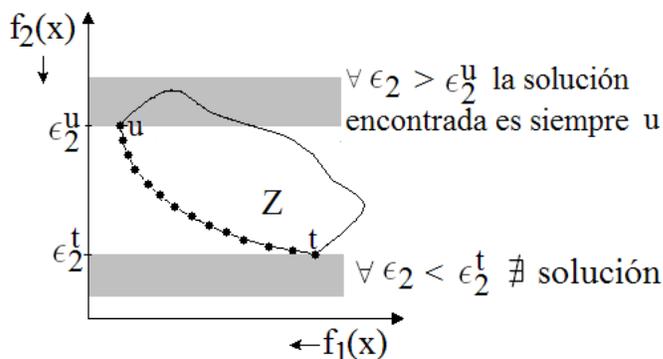


Figura 2.13: Ilustración gráfica de inconvenientes del método de las ϵ -restricciones.

de metas o indicando la relativa importancia de los diferentes objetivos son la programación por metas [25][26][102][103],[124], el ordenamiento lexicográfico [36][142], el método de Keeney y Raiffa [110], physical programming [138],... Remarcamos que los ejemplos (pesos, distancia a un punto de referencia y ϵ -restricciones) expuestos en la sección precedente, si los valores de los pesos w_i y de las restricciones ϵ expresan preferencias del decisor y por tanto no son utilizados como simples parametros, entonces se incluirían en esta sección. Los métodos de programación por metas y ordenamiento lexicográfico son a continuación presentados.

2.5.1 Programación por metas

Los orígenes de la programación por metas (goal programming en inglés) se sitúa en el artículo de Charnes et al. [26] en el año 1955. Se trata de un método de gran popularidad, con una gran cantidad de trabajos publicados tanto en aspectos teóricos como en aplicaciones a diferentes áreas. El proceso sigue los pasos siguientes:

1. Se formula un vector inicial \vec{T} de metas, que corresponde al nivel de logro que un decisor desea alcanzar para cada objetivo.
2. A cada objetivo $f_i(x)$ con $i \in [1, 2, \dots, k]$, se asocia dos nuevas variables denominadas: (i) de desviación positiva d_i^+ , la cual cuantifica el exceso de logro de una meta, (ii) de desviación negativa d_i^- , que cuantifica la falta de logro de una meta. Ambas variables deben respetar:

- d_i^+ y $d_i^- \geq 0$
- $d_i^+ \cdot d_i^- = 0$

3. Se minimizan ambas o una de las dos variables de desviación asociadas a cada objetivo. Cuando la meta proviene de un objetivo a minimizar, será la variable de desviación positiva d_i^+ la que se minimice. Por el contrario, si la meta proviene de un objetivo a maximizar será la variable de desviación negativa d_i^- la que se minimice. Por último, para una función objetivo cuya meta se quiere alcanzar exactamente, entonces se minimizan ambas variables de desviación d_i^+ y d_i^- . De esta manera el POM (2.3.1) es reformulado para obtener el problema:

$$\left\{ \begin{array}{l} \text{mín.} \quad (d_1^+ \text{ o } d_1^-, d_2^+ \text{ o } d_2^-, \dots, d_k^+ \text{ o } d_k^-) \\ \text{s.a.} \\ f_1(x) + d_1^- - d_1^+ = T_1 \\ \quad \quad \quad \cdot \\ \quad \quad \quad \cdot \\ f_k(x) + d_k^- - d_k^+ = T_k \\ g_j(x) \leq 0 \quad \quad \quad j \in (1, 2, \dots, m) \\ h_q(x) = 0 \quad \quad \quad q \in (1, 2, \dots, p) \\ B_{\text{inf}} \leq x_l \leq B_{\text{sup}} \quad \quad l \in (1, 2, \dots, n) \end{array} \right. \quad (2.5.1)$$

Minimización de las variables de desviación

El problema (2.5.1) muestra que el método de programación por metas consiste en la minimización de un vector (formado por las variables de desviación). La minimización de este vector, se puede afrontar de varias maneras:

1. Minimizar la suma de las variables de desviación. Por ejemplo:

$$\text{mín.} \quad (d_1^+, d_2^-, d_3^+, d_3^-, d_4^-) \quad (2.5.2)$$

tendremos que minimizar la siguiente suma:

$$\text{mín.} \quad (d_1^+ + d_2^- + d_3^+ + d_3^- + d_4^-)$$

2. Minimizar la suma ponderada de las desviaciones. Ahora se permite que un decisor asigne diferentes importancias, a la realización de las diferentes metas. Así, el problema (2.5.2) es minimizado según la siguiente suma:

$$\text{mín.} \quad (w_1 \cdot d_1^+ + w_2 \cdot d_2^- + w_3 \cdot (d_3^+ + d_3^-) + w_4 \cdot d_4^-)$$

3. Minimizar secuencialmente las componentes del vector. Primero, el decisor clasifica las metas en orden de prioridad (i.e. $P_1 \succ P_2 \succ \dots \succ P_k$). Luego, se

resuelve una secuencia de programas de forma que el primer programa minimiza la primera componente del vector que tiene asociada la mayor prioridad P_1 . El segundo programa minimiza la componente del vector que tiene asociada la prioridad P_2 , de forma que no se degrade el valor de la variable de desviación obtenida para la meta con prioridad P_1 . El proceso continua hasta que se resuelve el último programa que minimiza la componente del vector que tiene asociada la prioridad más baja P_k , sin degradar los valores de las variables de desviación obtenidas para las metas con prioridades más altas. Por ejemplo, si en el problema (2.5.2) clasificamos las metas según el orden de prioridad $T_2 = P_1 \succ T_3 = P_2 \succ T_1 = P_3 \succ T_4 = P_4$, el método resolverá la siguiente secuencia de programas:

- (a) minimizar d_2^- .
- (b) minimizar $d_3^+ + d_3^-$ considerando el valor de d_2^- constante.
- (c) minimizar d_1^+ considerando los valores de d_2^-, d_3^+ y d_3^- constantes.
- (d) minimizar d_4^+ considerando los valores de d_2^-, d_3^+, d_3^- y d_1^+ constantes.

Ejemplo

Formulemos el problema F_2 de Shaffer (2.4.6) como un modelo de programación por metas. Para ello vamos a asociar a la función objetivo $f_1(x)$ una meta de valor $T_1=2$ y a la función objetivo $f_2(x)$ una meta de valor $T_2=0.2$. A continuación se determina que variables de desviación son a minimizar. Para la función objetivo $f_1(x)$ las variables de desviación son d_1^+ y d_1^- y como $f_1(x)$ es a minimizar, es la variable d_1^+ la que se minimiza. De igual forma, a la función objetivo $f_2(x)$ se le asocia las variables de desviación d_2^+ y d_2^- y como $f_2(x)$ es a minimizar, se minimiza la variable d_2^+ . Si se considera que el decisor asigna la misma importancia a las diferentes metas, se obtiene el siguiente problema:

$$\left\{ \begin{array}{l} \text{mín. } d_1^+ + d_2^+ \\ \text{s.a.} \\ x^2 + d_1^- - d_1^+ = 2 = T_1 \\ (x - 2)^2 + d_2^- - d_2^+ = 0.2 = T_2 \\ x \in [0, 2] \end{array} \right. \quad (2.5.3)$$

resolviendo (2.5.3), se obtiene la siguiente solución óptima:

$$\begin{aligned} x &= 1.4142 \\ d_1^+ &= d_1^- = 0 \\ d_2^+ &= 0.1431, d_2^- = 0 \end{aligned}$$

Esta solución, realiza completamente la meta T_1 debido a que ambas variables de desviación d_1^+ y d_1^- toman el valor cero. Sin embargo, la meta T_2 es sobrepasada en el valor 0.1431 pues la variable de desviación que se desea minimizar obtiene el valor $d_2^+=0.1431$.

Discusión

Este método, genera una solución eficiente pero no necesariamente Pareto óptima. Además, en ciertas situaciones el método puede no ser capaz de encontrar soluciones que pertenezcan a zonas no convexas.

2.5.2 Ordenamiento lexicográfico

Este método resuelve una secuencia de problemas uniobjetivos (ver [36][142]), de forma que cada problema considera una sola función objetivo sujeto tanto a las restricciones del problema, como a los valores de las soluciones obtenidas en la resolución de los problemas anteriores. Si se parte del POM (2.3.1), se resuelven k problemas de la siguiente manera:

1.

$$\left\{ \begin{array}{ll} \text{mín. } f_1(x) & \\ \text{s.a.} & \\ g_j(x) \leq 0 & j \in (1, 2, \dots, m) \\ h_q(x) = 0 & q \in (1, 2, \dots, p) \\ B_{\text{inf}} \leq x_l \leq B_{\text{sup}} & l \in (1, 2, \dots, n) \end{array} \right.$$

Si se supone como solución del problema el valor $f_1(x) = S_1^*$, se resuelve a continuación el siguiente problema

2.

$$\left\{ \begin{array}{ll} \text{mín. } f_2(x) & \\ \text{s.a.} & \\ f_1(x) = S_1^* & \\ g_j(x) \leq 0 & j \in (1, 2, \dots, m) \\ h_q(x) = 0 & q \in (1, 2, \dots, p) \\ B_{\text{inf}} \leq x_l \leq B_{\text{sup}} & l \in (1, 2, \dots, n) \end{array} \right.$$

El procedimiento se repite hasta la función objetivo k . Terminándose con la resolución del problema:

3.

$$\left\{ \begin{array}{l} \text{mín. } f_k(x) \\ \text{s.a.} \\ f_1(x) = S_1^*, f_2(x) = S_2^*, \dots, f_{k-1}(x) = S_{k-1}^* \\ g_j(x) \leq 0 \quad j \in (1, 2, \dots, m) \\ h_q(x) = 0 \quad q \in (1, 2, \dots, p) \\ B_{\text{inf}} \leq x_l \leq B_{\text{sup}} \quad l \in (1, 2, \dots, n) \end{array} \right.$$

Discusión

El método es muy sencillo e intuitivo. Sin embargo dos ordenamientos diferentes de las funciones objetivo, pueden no obtener la misma solución. Lo que requiere por tanto, que el decisor realice un ordenamiento previo de la secuencia de objetivos a minimizar.

Capítulo 3

Algoritmos Evolutivos

3.1 Introducción

En “*The Origin of Species by Means of Natural Selection Or the Preservation of Favoured Races in the Struggle for Life*” (*El Origen de las Especies*), Darwin [44] expone su *Principio de la Evolución Natural*: los individuos mejor adaptados al medio ambiente, tienden a sobrevivir más largo tiempo y reproducirse más fácilmente, transmitiendo su información genética a sus sucesores. Según Darwin, una especie que no sufre cambios se vuelve incompatible con su ambiente, pues éste tiende a cambiar con el tiempo. Los parecidos entre padres e hijos le hicieron concluir que los cambios son heredados y transmitidos. En su teoría, la herencia juega un papel preponderante, sin embargo, desconocedor de las leyes de la herencia propuestas por Mendel [132], defendió la teoría de la *pangénesis* [43]; según esta teoría, el cuerpo produce pequeñas partículas hereditarias llamadas pangenes o gémulas, que son transportadas por la sangre. Esto significaba que las características de los padres se transmitían directamente a la sangre de los hijos.

Contemporáneo de Darwin, el monje austriaco Johann Gregor Mendel, publica en 1865 un trabajo en el que se resumían los experimentos que había realizado con plantas de guisantes. El documento original, publicado en alemán [132] no se tradujo al inglés [133] hasta 1901. Como resultado de sus experimentos, Mendel estableció tres leyes fundamentales que gobernaban el paso de una característica de un miembro de una especie a otro. La primera ley (de segregación) establece que cada par de alelos de un gen recibidos de los padres, se separan cuando se forman los gametos durante la meiosis. La segunda ley (de independencia) sostiene que los pares de alelos se independizan durante la formación de los gametos. La tercera ley (de uniformidad) establece que cada característica heredada se determina mediante dos factores provenientes de ambos padres, lo cual decide si determinado gen es dominante ó recesivo.

Treinta años más tarde de la publicación de *El Origen de las Especies* de Darwin, el

alemán August Weismann, estableció su teoría del *germoplasma*, [205], para ello cortó la cola a un grupo de ratas durante 22 generaciones, comprobando que la cola de las nuevas generaciones de ratas no se veía afectada en su longitud. En su teoría afirma que el cuerpo se divide en células germinales (*germoplasma*) que pueden transmitir información hereditaria y en células somáticas (*somatoplasma*) que no pueden hacerlo. Weismann llegó a la conclusión que la selección natural era lo único que podía cambiar el *germoplasma* o *genotipo* (composición genética de un individuo) y éste y el ambiente podían influenciar al *somatoplasma* o *fenotipo* (rasgos específicos de un individuo).

En la actualidad, la *teoría evolutiva*, considerada como combinación de las ideas de Darwin y complementada con las de Weismann y Mendel se conoce como el paradigma *Neo-darwiniano*. Este paradigma, sostiene que los procesos de la vida de nuestro planeta, se pueden explicar mediante unos pocos procesos estadísticos que actúan en las poblaciones y especies: la *reproducción*, la *mutación*, la *competencia* y la *selección*. El conjunto de estos procesos y su resultado, es la *evolución*. Una descripción detallada sobre la evolución de los seres vivos se puede consultar en C.A. Coello [33].

Inspirados en la *teoría evolutiva*, los *Algoritmos Evolutivos*¹ (AE) tratan de simular la evolución de una población de individuos (inicialmente generada de forma aleatoria) a la cual se aplican diferentes operadores (mutación, cruce...) y es sometida en cada generación a una selección. Los AE, utilizados para resolver problemas de optimización, constituyen un enfoque original, que sin necesitar conocimiento alguno del problema con el que tratan, logran encontrar no una solución exacta, pero sí una solución satisfactoria.

3.2 De los modelos biológicos a los programas informáticos

Las primeras tentativas de implementar los principios del Darwinismo en programas informáticos se sitúan en los años 1950. Desde entonces, versiones diferentes de AE han sido introducidas por diversos autores.

En 1954, Nils Aall Barricelli [10] desarrolla las primeras simulaciones de procesos de evolución y las utiliza sobre problemas de optimización.

George J. Friedman [77] propone en 1956, una aplicación de técnicas evolutivas a la robótica. En un proceso similar a la selección natural, propuso evolucionar circuitos de control, similares a lo que hoy se conoce por redes neuronales.

En 1958, R.M. Friedberg [75][76] con la idea de realizar cálculos sencillos, intenta evolucionar programas de ordenador generando conjuntos de instrucciones en lenguaje máquina. De forma informal, Friedberg fue el primero en utilizar los conceptos de

¹También Computación Evolutiva (del inglés Evolutionary Computation).

paralelismo implícito y esquemas. En el mismo año, Hans Joachim Bremermann [22], consideró la evolución como un proceso de optimización y usó cadenas binarias que se combinaban por medio de reproducción, selección y mutación.

Ingo Rechenberg [154] en 1965, presenta el primer algoritmo utilizando *Estrategias de Evolución*. En su algoritmo Rechenberg utilizó mutación, y la estrategia operativa fue un único individuo (no aún una población).

En el año 1966, Lawrence J. Fogel [66] presentó la *Programación Evolutiva*, la cual evolucionaba autómatas de estado finito, empleando un operador similar al de mutación.

En 1975, John H. Holland [98] propone los primeros *Algoritmos Genéticos*, utilizando tres operadores evolutivos fundamentales: selección cruce y mutación.

En el año 1985, se organiza la primera conferencia [88] sobre *Algoritmos Genéticos* en la Carnegie Mellon Pittsburgh University (USA).

En 1989, David E. Goldberg [85] publica uno de los libros más conocidos sobre *Algoritmos Genéticos*.

El término algoritmo memético [143] aparece en 1989.

En 1989, John R. Koza [118] propone la *Programación Genética*, la cual consiste en hacer evolucionar poblaciones de programas.

En 1993, el término *Evolutionary Computation* se expande con la aparición de la revista del mismo nombre, publicada por el Massachusetts Institute of Technology.

En el año 1997, Rainer Storn y Kenneth Price [187] presentan un algoritmo con Evolución Diferencial.

Como se aprecia, inspirados en los modelos biológicos, muchas son las técnicas y métodos que evolucionan programas informáticos. La mayor parte de las técnicas están descritas en Coello [33].

3.3 Paradigmas de los Algoritmos Evolutivos.

Históricamente, tres paradigmas desarrollados independientemente, conforman los AE: las *Estrategias Evolutivas* (EEs), la *Programación Evolutiva* (PE) y los *Algoritmos Genéticos* (AGs). En ocasiones, algunos autores añaden uno más: la *Programación Genética* (PG). Hoy en día, una literatura extraordinariamente abundante se puede encontrar sobre AE, además de ser considerados como un dominio de investigación muy prolífico y de enorme interés. A continuación se describen los cuatro paradigmas.

Estrategias Evolutivas.

Fué Rechenberg [154] quien introdujo la primera versión de EEs, concebida para la resolución de problemas de optimización con parámetros con valores reales. La versión original, creaba un solo hijo a partir de un solo padre, ambos rivalizaban por sobrevivir y el mejor de los dos se mantenía en la población mientras que el otro era eliminado. Este tipo de estrategia era denominada (1+1)-EE y se caracterizaba por la ausencia del operador de cruce, y la utilización de la mutación Gaussiana para asegurar la evolución. Rechenberg propuso la regla 1/5 para la adaptación de forma determinística de la desviación estándar de la mutación (La razón entre mutaciones exitosas y el total de mutaciones ha de ser 1/5. Si es mayor, entonces se incrementa la desviación estándar. Si es menor se decrementa). De otra parte, Bäck [5][7] propuso una regla de éxito 1/7.

Esta primera versión de EE, sirvió de base a Schwefel [177][178], para formular las estrategias $(\mu + \lambda)$ -EE y (μ, λ) -EE (donde μ son los padres y λ los hijos), añadiendo además el operador de cruce tomado de los AGs. En la primera estrategia, Schwefel selecciona los mejores individuos entre los padres y los hijos, mientras que en la segunda, selecciona los mejores entre la población de hijos. En las EEs, no sólo evolucionan las variables del problema sino también los parámetros propios de la técnica (las desviaciones estándar) lo que se conoce como *auto-adaptación*.

Programación Evolutiva.

La PE propuesta por L.J. Fogel [66], consiste básicamente en hacer evolucionar una población de autómatas de estado finito, y es utilizada para la resolución de problemas de predicción. La técnica consiste en exponer una población de autómatas de estado finito padres, a una secuencia de símbolos (ambiente). A medida que cada símbolo de entrada es presentado a los autómatas padres, se hace una predicción para determinar el símbolo de salida, el cual es comparado con el siguiente símbolo de entrada. La idea es que el autómata, pueda predecir la secuencia futura de símbolos de entrada. La evaluación de la aptitud de los individuos corresponde al número de símbolos predichos correctamente. Cada autómata padre de la población genera un hijo por mutación y los mejores individuos de entre padres e hijos son seleccionados para sobrevivir y pasar a la siguiente generación.

La PE ha sido extendida por David. B. Fogel [64][65], para que puedan trabajar con variables reales, y donde la selección determinista es reemplazada por un torneo estocástico.

Algoritmos Genéticos.

Los AGs son probablemente la rama de los Algoritmos Evolutivos más conocida y utilizada en la resolución de problemas de búsqueda y optimización. Fueron desarrollados por John Holland [97] a principios de los años 1960. La idea de su sistema era estudiar y diseñar, sistemas artificiales (o programas) que imitaran los procesos de adaptación de los sistemas naturales [97][98]. Los AGs, fueron aplicados por primera vez por De Jong [58] en el año 1975 a un problema de optimización paramétrica, aunque es a partir de finales de los 80, cuando se popularizan con la aparición del libro de referencia escrito por Golberg [85]. Un AG, se caracteriza por el hecho de hacer evolucionar poblaciones de individuos, aplicando los mismos métodos de la evolución biológica: selección de individuos, reproducción sexual y mutación.

Básicamente el funcionamiento de un AG sigue el pseudocódigo mostrado en el algoritmo 1. Inicialmente se genera una población aleatoria de individuos $P(0)$. En cada generación t , se evalúa la aptitud de todos los individuos que componen la población $P(t)$. A continuación y en base a la aptitud, se selecciona de $P(t)$ un grupo $P'(t)$ de padres (llamado grupo de reproducción) para la creación de descendientes. Luego, aplicando algún operador de cruce (o recombinación), los genes de los individuos padres de $P'(t)$ son cruzados para obtener la población de descendientes $P''(t)$. Por último, mediante el operador mutación los genes de los individuos de la población $P''(t)$ son perturbados y se obtiene la población $P'''(t)$. Nuevas poblaciones de individuos van a continuación a sucederse, hasta alcanzar el criterio de parada t_{\max} . Los AGs, a diferencia de la PE o las EEs, actúan sobre conjuntos de individuos o soluciones (genotipo) codificadas en cromosomas cuya representación tradicional es la binaria, y no sobre el fenotipo (la evaluación de un genotipo particular). En general no es posible asegurar la convergencia de los AGs al conjunto de soluciones óptimas y se demuestra [168], que los AGs requieren de elitismo (mantener intacto al mejor individuo de la población en cada generación) afín de converger al óptimo.

Programación Genética.

Desarrollada por John R. Koza [118] [119], la PG evoluciona poblaciones de individuos que representan programas o autómatas, descritos mediante S-expresiones Lisp, las cuales tienen la ventaja de ser fácilmente representadas en árboles. Inicialmente, se genera una población aleatoria de programas y en cada iteración del algoritmo se evalúa la aptitud de cada programa. Luego se realiza una selección en base a la aptitud y se aplican los operadores de cruce y mutación para la creación de una nueva población. El principio es el mismo que el de un AG clásico, pero los operadores de cruce y mutación son algo diferentes pues trabajan directamente sobre la estructura de árbol del programa. La PG ha sido utilizada en la resolución de un gran número de

problemas cuyas soluciones pueden ser representadas por estructuras arborescentes, como funciones lineales [147], planeación de movimientos de robots [119], grafos [194], entre otros.

Algoritmo 1. *Pseudocódigo básico de un AG*

```

 $t_{\max};$            //Numero máximo de generaciones.
 $t=0;$ 
Generación de  $P(t);$  //se genera aleatoriamente la población  $P(0)$ .
Mientras  $t \leq t_{\max};$  //test para criterio de parada.
  Evaluación;       //se evalúa la aptitud de cada individuo de  $P(t)$ .
  Selección;        //basándose en la aptitud calculada en la fase anterior,
                    //se selecciona de  $P(t)$  una subpoblación  $P'(t)$  de padres
                    //para la producción de descendientes.
  Cruce;            //se cruzan los genes de los padres de  $P'(t)$  para
                    //obtener  $P''(t)$ .
  Mutación;         //se perturban de forma estocástica los genes de los
                    //individuos de  $P''(t)$  para obtener  $P'''(t)$ .
 $P(t+1)=P'''(t);$ 
 $t=t+1;$ 
Fin Mientras;

```

Comparación de paradigmas.

En los trabajos de Coello [33] y S. Esther Alonso [2], se hace una interesante comparativa entre paradigmas. La comparativa se muestra en la tabla 3.1 (tomada de [2] y donde la autora remarca tener en cuenta que la información se refiere a las versiones originales de los paradigmas, y por tanto existen variaciones en posteriores versiones como por ejemplo: los AG actuales también pueden trabajar en codificación real).

3.4 Terminología.

- Cromosoma. Codificación de una solución² del problema a partir de un conjunto de genes (figura 3.1). En general se trata de una cadena con valores binarios, pero también pueden ser cadenas con valores enteros o reales.
- Gen. Una subcadena del cromosoma que codifica el valor de una variable x_i (figura 3.2).
- Alelo. Los distintos valores que conforman un gen. En caso de codificación binaria el valor de un alelo es 0 o 1 (figura 3.2).
- Locus. Define la posición de un gen dentro del cromosoma (figura 3.2).

²En los AE se utilizan indistintamente los términos solución, individuo y cromosoma.

Tabla 3.1: Tabla comparativa entre paradigmas

	PE	AG	EE	PG
Creador de la técnica	Fogel	Holland y Golberg	Rechenberg y Schwefel	Koza
Representación	Real	Binaria	Real (vector de variables, desviaciones estándar y ángulos de rotación)	S-expresión de Lisp en forma de árboles
Auto-adaptación	Ninguna salvo en algunas versiones donde se adaptan varianzas o coeficientes de correlación	Ninguna	Desviaciones estándar y ángulos de rotación	Ninguna
Selección	Probabilística, extintiva	Probabilística, basada en la preservación	Determinística, extintiva o basada en la preservación	Probabilística
Cruce	Ninguno	Cruce de dos puntos, uniforme...	Discreta e intermedia, sexual y panmítica	Operador primario
Mutación	Gaussiano (operador único)	Inversión de bits (operador secundario)	Gaussiano (operador principal)	Operador secundario
Nivel al que trabajan	Especie	Individuo	Individuo	Individuo
Nivel estructural	Fenotípico	Genotípico	Fenotípico	Fenotípico

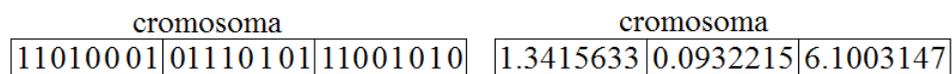


Figura 3.1: Ilustración de un cromosoma binario (izquierda) y un cromosoma real (derecha).

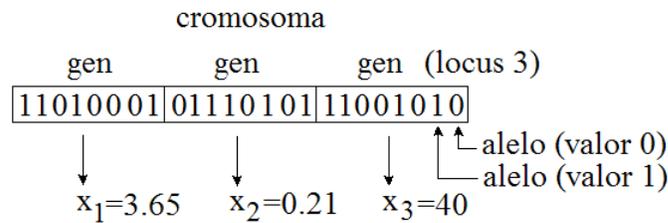


Figura 3.2: Ilustración de un cromosoma binario de tres genes que codifican las variables x_1 , x_2 y x_3 .

- Genotipo³. La codificación (por ejemplo binaria) de las variables que representan la solución de un problema (figura 3.3).
- Fenotipo³. Valores de la decodificación de un genotipo particular (figura 3.3).
- Individuo. Una solución del espacio de búsqueda, representada por un cromosoma.
- Superindividuo. Cromosoma que tiene un valor de adaptación muy alto en comparación con el resto de los cromosomas de la población.
- Población. Conjunto finito de individuos.
- Aptitud. Valor que se le asigna a cada individuo de una población, y que indica que tan bueno es con respecto a los demás para la solución de un problema. Se calcula mediante la función de aptitud, la cual, en ocasiones puede ser igual al valor de la función objetivo del problema.
- Generación⁴ (t). Conjunto de operaciones que permiten pasar de una población $P(t)$ a otra $P(t+1)$.
- Elitismo. Mecanismo utilizado por algunos AE, que copia a los mejores individuos de la población en una generación, en la población siguiente.
- Selección. Proceso de elección de individuos para la reproducción basada en la aptitud.
- Cruce. Operador de reproducción aplicado con la probabilidad p_c , y que forma uno o varios nuevos cromosomas, combinando partes de cada uno de sus cromosomas padres.
- Mutación. Operador que aplicado con una probabilidad p_m , forma un nuevo cromosoma a través de alteraciones de los valores de los genes de un solo cromosoma.

³Es importante no confundir genotipo y fenotipo: dos individuos con diferentes genotipos pueden tener exactamente el mismo fenotipo.

⁴También se utiliza el término iteración.

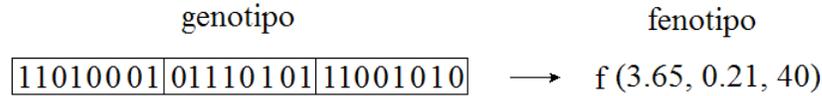


Figura 3.3: Ilustración de la decodificación del genotipo al fenotipo.

3.5 Codificación de variables

Desde un punto de vista informático, los AEs requieren que los genes (variables del problema) que corresponden a cada cromosoma, sean codificados en una cadena o ristra de símbolos. La codificación más utilizada es la binaria (cadena de bits 0-1). Holland [98] dio una justificación teórica para usar este tipo de codificación. Según Holland, la codificación binaria cuenta con más esquemas⁵, lo que favorece la diversidad e incrementa la posibilidad de que se formen buenos bloques⁶ constructores. Sin embargo, las ventajas de la codificación binaria frente a otras, ha sido cuestionada. C.A. Coello [33] apunta que, en la resolución de ciertos problemas del mundo real, si se trata de optimizar una función con alta dimensionalidad (por ejemplo 50 variables) y se quiere trabajar con alta precisión (por ejemplo 5 decimales), se generan cadenas muy largas (del orden de 1000 bits), y el AG tendrá muchos problemas para producir resultados aceptables en la mayoría de las veces.

Una de las ventajas de la codificación binaria, es que se puede fácilmente codificar valores reales, enteros, booleanos, cadenas de caracteres etc. Para ello, simplemente se necesita calcular la longitud de la cadena binaria y usar las funciones de codificación y decodificación, tal y como se describe más abajo para el caso de valores reales.

Existen otras alternativas a la codificación binaria. Por ejemplo, la codificación real fue introducida inicialmente para las EEs y se extendió rápidamente a otros tipos de AEs. John R. Koza [118], utiliza una representación de árbol para el desarrollo de la PG.

Calculo de la longitud de la cadena binaria

Para variables reales, la longitud de la cadena binaria depende de la precisión que se desee, así por ejemplo, si el intervalo de la variable x_i es $[a_i, b_i]$ y la precisión requerida es de n dígitos después del punto decimal, entonces la longitud del intervalo debe ser discretizada en al menos $(a_i - b_i) * 10^n$ puntos. En términos binarios, la

⁵Cadena formada por la terna de símbolos *,0,1 (usando alfabeto binario), donde el símbolo * es un comodín que puede valer un 0 o un 1.

⁶Porción de un cromosoma, que da una probabilidad alta de incrementar la aptitud a la cadena cromosómica en la cual está presente.

cadena de bits m_i requeridos será:

$$2^{m_i-1} < (b_i - a_i) * 10^n \leq 2^{m_i} - 1. \quad (3.5.1)$$

Decodificación

La ecuación a utilizar para la decodificación es la siguiente:

$$x_i = a_i + g_i \left(\frac{b_i - a_i}{2^{m_i} - 1} \right) \quad (3.5.2)$$

donde $g_i = \sum_{j=0}^{m_i-1} 2^j$ convierte un string binario de la base 2 a la base 10.

Codificación

Para la codificación se utiliza la ecuación:

$$g_i = \left(\frac{x_i - a_i}{b_i - a_i} \right) * (2^{m_i} - 1) \quad (3.5.3)$$

Ejemplo

Supongamos ahora, un problema en el que queremos codificar en binario la variable definida en el dominio $x_i \in [-1, 2]$ y de precisión requerida 6 dígitos.

Longitud de la cadena binaria

Necesitaremos una discretización de 3.000.000 puntos para cubrir todo el intervalo, pues: $3 * 10^6 = 3.000.000$. En términos binarios requerimos una cadena de $m_i = 22$ bits, ya que según (3.5.1) se tiene que: $2.097.152 = 2^{21} < 3.000.000 \leq 2^{22} - 1 = 4.194.303$. Los extremos del intervalo -1 y 2, se representan respectivamente por las cadenas (cromosomas):

```
00000000000000000000000000000000
11111111111111111111111111111111
```

Decodificación

Supongamos ahora un cromosoma del problema dado, por ejemplo:

```
1000101110110101000111
```

aplicando las ecuaciones de decodificación (3.5.2) obtenemos:

$$g_i = 2^0 + 2^1 + \dots + 2^{22} = 2.288.967$$

$$x_i = -1 + 2.288.967 \left(\frac{3}{4.194.303} \right) = 0.637197$$

Codificación

Aplicando la ecuación de codificación (3.5.3) se obtiene:

$$g_i = \left(\frac{0.637197 + 1}{3} \right) * 4.194.303 = 2.288.967$$

3.6 Operadores evolutivos básicos

Existe una amplia variedad de operadores evolutivos que se utilizan en AEs. De entre ellos destacan tres operadores básicos, que son los que llevan a cabo la evolución poblacional: el operador selección, el operador cruce y el operador mutación. Estos tres operadores se describen a continuación.

3.6.1 Operadores de selección.

El operador selección, permite identificar estadísticamente los mejores individuos de una población y eliminar parcialmente los malos. Sin embargo, como sucede en la naturaleza, no porque un individuo es bueno sobrevivirá necesariamente, y de la misma manera, no porque es malo debe desaparecer. En efecto, a veces, una especie bien adaptada puede descender de un individuo calificado malo. Varios métodos son posibles, analizaremos a continuación algunos de ellos.

Selección proporcional.

Propuesta en 1975 por Holland [98], los individuos se eligen acorde a su contribución aptitud con respecto a la población total. Algunos ejemplos de este tipo de selección son:

1. *Selección de Ruleta (Roulette Wheel Selection RWS)*. Fue propuesta por De Jong [58] y popularizada por Goldberg [85]. Se trata de un método estocástico que simula una ruleta de casino (figura 3.4 izquierda) con tantas casillas como

individuos de la población. Las casillas tienen un tamaño proporcional a la probabilidad asignada a cada individuo x_i dada por la expresión:

$$p(x_i) = \frac{f(x_i)}{\sum_{i=1}^N f(x_i)}$$

donde $f(x_i)$ es el valor aptitud del individuo x_i , N el número de individuos de

la población y $\sum_{i=1}^N f(x_i) = 1$.

La ruleta es lanzada cada vez que es escogido un individuo.

El valor esperado v_e del número de copias que se esperan obtener de un individuo x_i de la población en la generación t , está dado por el cociente entre el valor de la aptitud del individuo y la media de los valores de las aptitudes de los individuos de la población:

$$v_e = \frac{N}{\sum_{i=1}^N f(x_i)} f(x_i)$$

Este método puede producir una pérdida de diversidad de la población si la presión de selección (cociente entre el valor de la aptitud máxima y la media de los valores de las aptitudes de los individuos de la población) es muy elevada.

2. *Selección Universal Estocástica (Stochastic Universal Selection SUS)*. Fue propuesta por Baker [8] con el fin de minimizar la mala distribución de los individuos en la población en función de sus valores esperados. Al igual que en la RWS, se considera que la ruleta tiene tantas casillas como individuos de la población y tamaño de las casillas proporcional a la probabilidad asignada a cada individuo. Pero ahora, la ruleta es lanzada una sola vez y los individuos son escogidos por N puntos equidistantes (figura 3.4 derecha).

Selección por torneo.

Este tipo de selección fue estudiada en su tesis por Brindle [23] y propuesta, por Wetzel [206]. La manera de operar, consiste en escoger de forma aleatoria k (tamaño

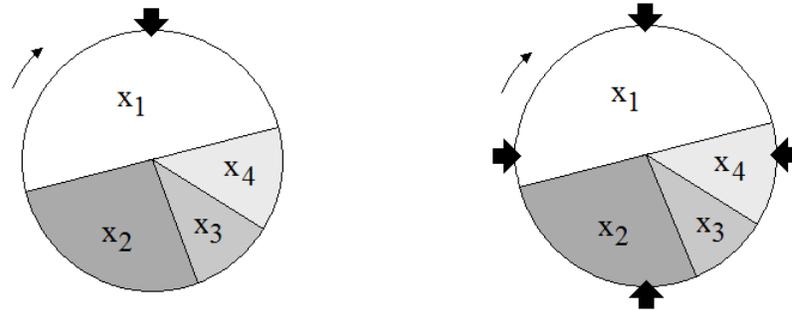


Figura 3.4: Selección RWS (izquierda): se lanza la ruleta N veces y cada vez se selecciona el individuo marcado. Selección SUS (derecha): se lanza la ruleta una sola vez y se seleccionan los N individuos marcados.

del torneo) individuos de la población, y seleccionar el de mayor función de adaptación en el grupo. El proceso se repite hasta obtener el número de individuos requeridos. La presión de selección es fácil de ajustar variando el número k de participantes en el torneo (una k elevada conduce a una fuerte presión de selección). Cuando $k=2$, la selección es denominada por torneo binario.

3.6.2 Operadores de cruce.

Este operador permite que grupos de cromosomas intercambien fragmentos de sus genotipos para dar nuevas cromosomas. Bajo esta perspectiva de ser un intercambiador de material genético, el operador de cruce va a permitir descubrir mejores individuos, combinando las ventajas de los individuos ya descubiertos. Lo habitual es que esta técnica se aplique sobre representaciones binarias, pero con las modificaciones apropiadas se pueden generalizar a otros alfabetos. El cruce se aplica de acuerdo al parámetro probabilidad de cruce p_c (se aplica el cruce si el número aleatorio generado es menor o igual a p_c y no en caso contrario). Habitualmente se toman valores comprendidos entre 0.6 y 1.0, es usual el valor $p_c = 0.8$. Varios tipos de cruce son posibles, de entre ellos los más utilizados son el cruce en un punto, el cruce en dos puntos y el cruce uniforme, los cuales son descritos a continuación.

Cruce de un punto.

Propuesto por Holland [98] en 1975, consiste en intercambiar cadenas binarias de longitud fija, según un punto de corte elegido al azar (figura 3.5). Hay que hacer notar que el cruce se efectúa directamente a nivel binario, y no a nivel de genes. Por tanto, un cromosoma puede ser cortado en medio de un gen.

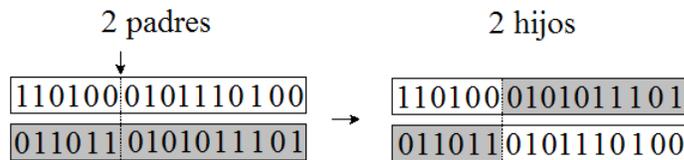


Figura 3.5: Representación esquemática del cruce de un punto.

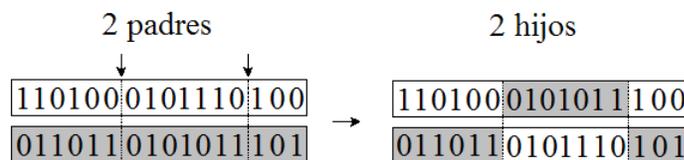


Figura 3.6: Representación esquemática del cruce de dos puntos.
Se suele aplicar una probabilidad de 0.5

Cruce de dos puntos.

Propuesto por DeJong [58]. Ahora, segmentos de cadenas binarias de longitud fija, son intercambiadas según dos puntos de corte elegidos al azar (figura 3.6). El cruce de dos puntos es generalmente considerado como más eficaz que el de un punto [33].

Cruce uniforme.

Fue propuesto por Syswerda [189]. Se puede ver como un cruce de n puntos, pero en el cual el número de puntos de cruce no se fija previamente, sino que es determinado de forma aleatoria en el transcurso de la operación. En la práctica, se genera una cadena intermedia o máscara de cruce (de la misma longitud que el cromosoma), que dice qué posiciones se intercambian entre sí. Un 1 en la cadena intermedia, indica que los bits representados en la misma posición de los progenitores se han de cruzar, mientras que un cero indicaría que no han de hacerlo. Se suele aplicar una probabilidad de 0.5 para generar los unos de la máscara de cruce. En la figura 3.7, los puntos 2,5,10,11,14 de la cadena intermedia indican las posiciones de los bits de los progenitores que se deben cruzar.

3.6.3 Operadores de mutación.

El operador mutación va a permitir aumentar la diversidad poblacional, aplicándose tras el cruce a cada uno de los descendientes. Consiste en la inversión de un bit en un

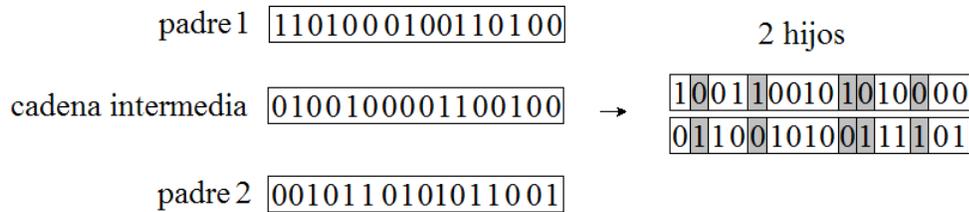


Figura 3.7: Representación esquemática del cruce uniforme.

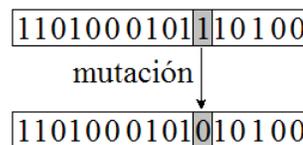


Figura 3.8: Representación esquemática de una mutación en un cromosoma.

cromosoma. De la misma manera que para el operador de cruce, se debe establecer un parámetro de probabilidad de mutación p_m (se aplica la mutación si el número aleatorio generado es menor o igual a p_m y no en caso contrario). Se recomiendan [33] valores de p_m entre 0.001 y 0.01, este último valor es bastante frecuente. Thomas Bäck [6] sugiere utilizar una $p_m = 1/L$, donde L es la longitud del cromosoma. Debe tenerse en cuenta, que la utilización de probabilidades de mutación muy elevadas, pueden convertir la búsqueda en aleatoria. Para representaciones binarias, muy común es la utilización de la mutación uniforme, ésta consiste en que cada alelo de la cadena cromosómica, se intercambia por su contrario (de 0 a 1 o viceversa) con una determinada probabilidad de mutación p_m (figura 3.8).

3.7 Otros operadores evolutivos.

Entre otros operadores que resultan de interés, destacamos los siguientes:

Elitismo.

El operador elitismo intenta asegurar que los individuos más aptos de una población en una generación, pasen a la siguiente sin ser modificados por ningún operador genético, asegurando que la aptitud máxima de la población no se reduzca de una generación a la siguiente.

Eliminación de duplicados.

En el transcurso de la evolución del AE, es posible que se generen individuos que no se quiere que se copien, por ya existir en la población. El operador de eliminación, lo evita comparando el nuevo individuo que se acaba de generar, con todos los de la población, si ya existe simplemente lo elimina, en caso contrario lo inserta en la población. El inconveniente de este operador es su alto costo computacional.

Tamaño de la población.

Se trata de un parámetro que a mayor tamaño, hace aumentar la diversidad de la población, mejorando con ello el desempeño del AE. Sin embargo, hay que hacer notar que su incremento requerirá un mayor tiempo de cálculo, ralentizando la evolución del AE. Contrariamente, una disminución del tamaño poblacional acelera la evolución, aunque si la disminución es excesiva, la calidad de la convergencia de soluciones puede ser mala.

Mecanismo de generación de la población inicial.

Se trata de un mecanismo, que pretende producir una población inicial de individuos, lo más diversa posible. En la figura 3.9, se observa que el caso a, al estar la población mejor distribuida, resulta más favorable para la búsqueda del óptimo que en el caso b. Si no se tiene ninguna idea acerca de la posición del óptimo en el espacio de búsqueda, se generan soluciones de forma aleatoria, realizando tiradas uniformes en todo el dominio del espacio de búsqueda, haciendo que los individuos respeten las restricciones del problema. Si por el contrario, se dispone de información a priori sobre el problema, que indica un sub-dominio donde es seguro que se encuentra el óptimo, entonces se generan individuos en ese sub-dominio, con la finalidad de acelerar la búsqueda.

Condición de parada.

Antes de ejecutar un AE, es imprescindible establecer cual es la condición de parada del algoritmo. Básicamente tres tipos de criterios de parada son conocidos:

1. Parada después de un número fijo de generaciones.
2. Parada desde que el valor del mejor individuo alcance un cierto umbral.
3. Parada después que un cierto número de generaciones se han realizado sin que la solución haya mejorado.

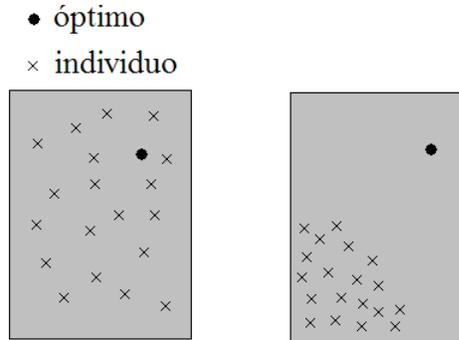


Figura 3.9: Generación de la población inicial: la figura izquierda en comparación con la figura derecha, muestra una situación más favorable para la búsqueda del óptimo.

Mantenimiento de diversidad.

Los operadores de mantenimiento de diversidad de individuos en la población de un AE, son mecanismos que se utilizan con la finalidad de evitar la convergencia prematura del algoritmo. Ejemplos conocidos de mantenimiento de diversidad son por ejemplo, el mecanismo de *superpoblación* [58], el *esclarecimiento elitista* [152] y la técnica muy popular introducida por Goldberg y Richardson [86] en 1987, conocida como *Función de Compartición* (Sharing Function Model) y que se describe a continuación.

Sharing Function Model

La idea que subyace en esta técnica, consiste en degradar el valor de la prestación del individuo o solución, en función de la densidad de la población en su vecindad, de esta manera se penalizan los individuos que están muy próximos entre si. El valor de la función aptitud compartida de un individuo i está dado por:

$$F'(i) = \frac{f(i)}{\sum_{j=1}^N Sh(d_{i,j})} \quad (3.7.1)$$

donde $d_{i,j}$ es la distancia entre las soluciones i -ésima y j -ésima y $Sh(d_{i,j})$ es la Función de Compartición. El valor de $Sh(d_{i,j})$, depende de la distancia entre dos soluciones y toma un valor comprendido entre $[0,1]$. Si $d_{i,j}$ es cero (lo que significa que dos soluciones son idénticas o su distancia es cero), entonces la Función de Compartición toma el valor $Sh(d_{i,j}) = 1$. Si $d_{i,j} \geq \sigma_{share}$ (lo que significa que la distancia entre

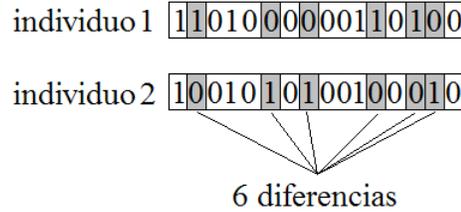


Figura 3.10: Cálculo de la distancia $d_{1,2}$ de Hamming en el espacio del genotipo.

dos soluciones supera un cierto umbral σ_{share}), entonces la Función de Compartición toma el valor $Sh(d_{i,j}) = 0$. Cualquier otro valor de distancia $d_{i,j}$ entre dos individuos tendrá un efecto de compartición parcial sobre cada una de ellas. La Función de Compartición más utilizada está dada por la expresión [86]:

$$Sh(d_{i,j}) = \begin{cases} 1 - \left(\frac{d_{i,j}}{\sigma_{share}}\right)^\alpha & \text{si } d_{i,j} \leq \sigma_{share} \\ 0 & \text{en otro caso} \end{cases} \quad (3.7.2)$$

donde α es una constante utilizada para controlar el aspecto de la Función de Compartición. Si se usa $\alpha=1$, el efecto de compartición se reduce linealmente desde uno a cero. Además, según como esté definida la distancia $d_{i,j}$ entre individuos, se pueden distinguir tres tipos de Funcion de Compartición:

1. Funcion de Compartición en el espacio de los individuos o espacio del genotipo. Por ejemplo, puede calcularse la distancia Hamming (en la figura 3.10 esta distancia vale $d_{1,2}=6$).
2. Funcion de Compartición en el espacio de la decisión (figura 3.11).
3. Funcion de Compartición en el espacio objetivos o espacio del fenotipo. Por ejemplo, se puede utilizar la expresión de distancia dada por:

$$d_{i,j} = \frac{f(i) - f(j)}{f^{\max} - f^{\min}} \quad (3.7.3)$$

donde $f(i)$ y $f(j)$ son los valores de las funciones objetivo para los individuos i y j . f^{\max} y f^{\min} son los valores máximos y mínimos de la función objetivo.

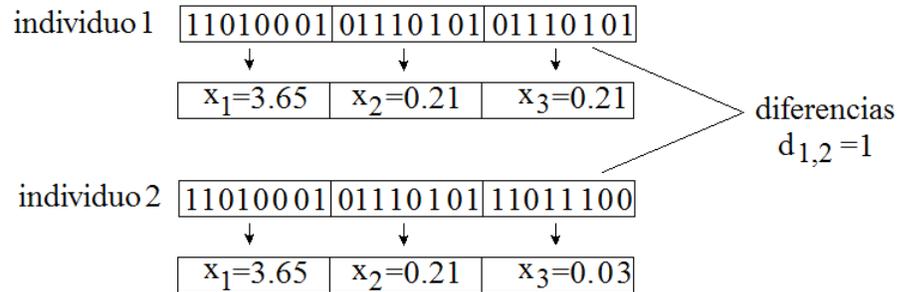


Figura 3.11: Cálculo de la distancia $d_{1,2}$ en el espacio de la decisión.

3.8 Equilibrio Explotación-Exploración

Para garantizar la eficacia de un AE, es preciso conseguir mantener un equilibrio entre la *explotación* de las buenas soluciones encontradas y la *exploración* de zonas desconocidas del espacio de búsqueda. Una excesiva explotación de las buenas soluciones, dificulta la exploración del espacio de búsqueda. Inversamente, el exceso de exploración obstaculiza claramente la explotación.

Dos factores esenciales que permiten abordar el equilibrio explotación-exploración en un AE son la *presión de selección* y la *diversidad poblacional* [207]. Ambos están inversamente relacionados: una alta presión de selección conlleva a una rápida pérdida de diversidad en la población, pues ésta se hace más homogénea y su evolución se reduce a la de los individuos dominantes; por el contrario, mantener diversidad en la población, permite introducir en ella, nueva información capaz de contrarrestar el exceso de presión de selección.

La mayor parte de los operadores y parámetros (tamaño de la población, tipo de selección, tipo de cruce, valores de las probabilidades de cruce y mutación etc.) utilizados en el ajuste de un AE, son en realidad términos indirectos de actuación sobre la presión de selección y la diversidad poblacional. La forma en que son definidos, contribuirá al éxito o fracaso del algoritmo.

Capítulo 4

Algoritmos Evolutivos Multiobjetivo. Estado del Arte

4.1 Introducción

Los Algoritmos Evolutivos Multiobjetivo (AEMOs), constituyen un tipo de metaheurística, que ha revelado un extraordinario potencial en la resolución de problemas optimización con múltiples objetivos. A diferencia de otros métodos de optimización (que en su mayoría) operan sobre una única solución, los AEMOs manipulan una población de soluciones. Esta diferencia les proporciona la habilidad de encontrar múltiples soluciones Pareto óptimas, en problemas multiobjetivo NP-difíciles con espacios de soluciones ya sean continuos, no continuos, discretos, no convexos, etc.

La implementación algorítmica básica de un AEMO sigue el pseudocódigo descrito por el algoritmo 2, el cual muestra que el funcionamiento de un AEMO difiere con el de un AE (ver en la sección 3.3 el algoritmo 1), en la forma de asignar la aptitud a los individuos de la población. Mientras en un AE se calcula un valor de aptitud para cada individuo (a partir de los valores de una sola función objetivo), un AEMO necesita de una fase adicional (fase de transformación), la cual transforma un vector (de componentes los valores de las funciones objetivo de cada individuo de la población) en un único valor aptitud.

Dos propiedades fundamentales [11][16][50][219], que se esperan encontrar en toda implementación evolutiva multiobjetivo son:

1. Convergencia hacia la frontera de Pareto. El conjunto de soluciones alcanzado, debe estar lo más cerca posible del verdadero frente de Pareto (figura 4.1 izquierda).
2. Diversidad de soluciones en la frontera de Pareto: El conjunto de soluciones encontrado, debe poseer una buena distribución y amplitud (figura 4.1 izquierda).

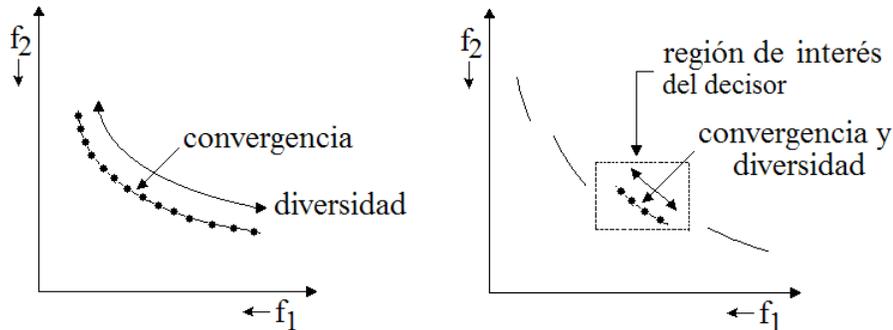


Figura 4.1: Ilustración de dos objetivos fundamentales de un AEMO: convergencia y diversidad.

También, algunos autores [1][54][115][219] anotan una tercera propiedad: que las dos primeras se alcancen en un tiempo de computación aceptable.

Algoritmo 2. Pseudocódigo básico de un AEMO

```

 $t_{\max}$ ; //Numero máximo de generaciones.
 $t=0$ ;
Generación de  $P(t)$ ; //se genera aleatoriamente la población  $P(0)$ .
Mientras  $t \leq t_{\max}$ ; //test para criterio de parada.
  Evaluación; //se calculan los valores de las funciones objetivo para
  cada individuo de  $P(t)$ .
  Transformación; //se transforma el vector que contiene los valores de las
  funciones objetivo para cada individuo (obtenido en
  la fase anterior), en una aptitud.
  Selección; //basándose en la aptitud calculada en la fase anterior,
  se selecciona de  $P(t)$  una subpoblación  $P'(t)$  de padres
  para la producción de descendientes.
  Cruce; //se cruzan los genes de los padres de  $P'(t)$  para
  obtener  $P''(t)$ .
  Mutación; //se perturban de forma estocástica los genes de los
  individuos de  $P''(t)$  para obtener  $P'''(t)$ .
 $P(t+1)=P'''(t)$ ;
 $t=t+1$ ;
Fin Mientras;

```

Cuando se trata de procesos de búsqueda multiobjetivo que manipulan a priori las preferencias del decisor, Salem F. Adra [1] señala la exigencia a cualquier implementación evolutiva, de converger y diversificar en la región de interés del decisor (figura 4.1 derecha).

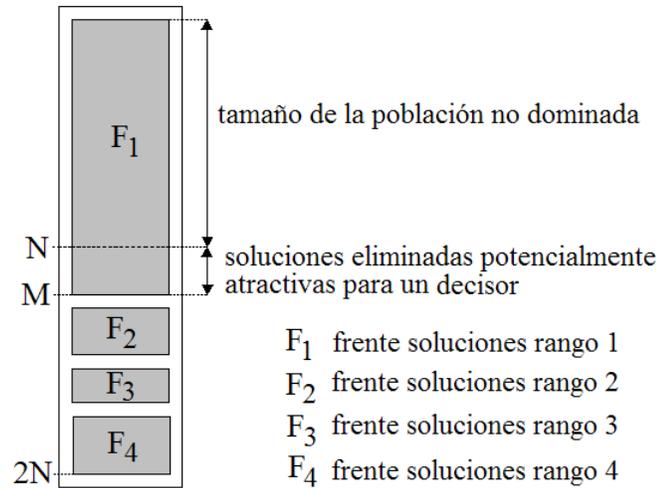


Figura 4.2: Ilustración de un primer inconveniente en problemas reales discretos: el tamaño N de la población no-dominada adoptado es inferior al tamaño M (desconocido) del frente de Pareto.

En cualquier caso, la idea que subyace es proporcionar al decisor un representativo conjunto de soluciones en términos del espacio de soluciones de los objetivos, para que elija una solución acorde a sus preferencias.

Sin embargo, es conveniente señalar que esta perspectiva pudiera no ser siempre la más apropiada. Por ejemplo, en muchos problemas reales discretos:

- El número de soluciones del frente de Pareto es en la mayoría de ocasiones desconocido. Si el tamaño especificado para la población de soluciones no dominadas del AEMO es menor que el número de soluciones del problema (figura 4.2), el frente de soluciones encontrado, estará incompleto, lo que obliga a que se pierdan soluciones que pudieran ser potencialmente atractivas para el decisor [135].
- Reconocidos AEMOs (por ejemplo NSGAII [55]), utilizan un mecanismo de evolución tal que a partir de una determinada generación toda la población pudiera estar incluida en el primer frente de soluciones no-dominadas. En esta situación, se pueden eliminar soluciones Pareto óptimas situadas en una región muy poblada y conservar soluciones no dominadas en la población en curso pero que no son Pareto óptimas [158] (figura 4.3).

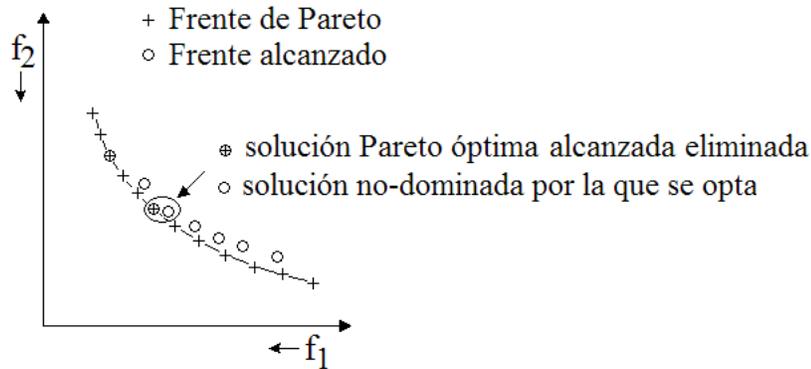


Figura 4.3: Ilustración de un segundo inconveniente en problemas reales discretos: en una región poblada, una solución Pareto óptima es eliminada, conservándose otra que pese a ser no-dominada en la actual generación no es Pareto óptima.

4.2 Clasificación de los AEMOs

Numerosos autores [11][29][31][70][184], clasifican los AEMOs en función de su forma de tratar las funciones objetivo en la fase de transformación (ver Algoritmo 2), ordenándolos en tres grupos:

1. **Métodos agregativos.** Se tratan de técnicas inspiradas en la optimización clásica. La más sencilla, combina (o agrega) todas las funciones objetivo del problema como una suma ponderada, de forma que la valoración de la función agregada (aptitud) es utilizada en la selección de los individuos de la población. En general las aplicaciones evolutivas que han hecho uso de esta técnica, utilizan los pesos, no como expresión de la importancia de los objetivos, sino como parámetros que varían para obtener soluciones Pareto óptimas. La principal desventaja de esta técnica es la imposibilidad de generar soluciones en frentes de Pareto no convexos. Métodos como RWGA¹ de Murata e Ishibuchi [144], HLGGA² de Hajela y Lin [94], son ejemplos de entre otros muchos [104][109][190][202].

Otra alternativa de agregación es el método de las ϵ -restricciones. La técnica consiste en optimizar una función f_k , mientras las otras funciones objetivo son tratadas como restricciones del problema. De esta forma, lo que se resuelve es un problema uniobjetivo (de función objetivo f_k) de forma que variando los

¹Random Weighted Genetic Algorithm.

²Hajela & Lin Genetic Algorithm.

valores de ϵ , se pueden generar soluciones Pareto óptimos. Algunos ejemplos que utilizan esta técnica son [125][126][156].

Una tercera técnica de agregación es la agregación por metas. Con este método, un decisor define las metas que desea alcanzar para cada objetivo. A continuación, Estos valores se introducen en la formulación del problema, transformándolo en un problema uniobjetivo. Entre otros AEMOs que han tratado este enfoque se tiene [49][173][208].

2. **Métodos no Pareto.** Estos métodos se basan en poblaciones de soluciones. Su principio consiste en un proceso de búsqueda de soluciones tratando separadamente las funciones objetivo (selección paralela). El principal valedor de este enfoque es el algoritmo VEGA³ de Shaffer [174][175]. La técnica ha sido utilizada por otros autores [108][188].

Otro enfoque también clásico, de los métodos no Pareto es la de la selección léxicográfica. El método puede tener un especial interés cuando existe una clara jerarquía entre las diferentes funciones objetivo. Así, un decisor define un orden de importancia de las funciones objetivo, a partir de la cual se realiza la selección de individuos. En [74][122] la selección se realiza comparando pares de individuos según la función objetivo con mayor prioridad, si el resultado fuese el mismo para ese objetivo, se utilizaría la siguiente función objetivo en prioridad, etc.

Por último, cabe mencionar otra aproximación basada en la técnica de juegos no-cooperativa según el concepto del equilibrio de Nash [145][146]. Periaux y Sefrioui [151][179] utilizan este concepto y proponen su aplicación en AEMOs, para tratar problemas de dinámica de fluidos e ingeniería aeronáutica, generando una solución no dominada (punto de equilibrio de Nash) en cada ejecución.

3. **Métodos de rangos de Pareto.** Contrariamente a los métodos anteriores, que utilizan en el proceso de selección de individuos una función agregada o bien tratan separadamente las diferentes funciones objetivo, los métodos basados en rangos de Pareto usan el concepto de dominancia, de forma que la población es clasificada en rangos acorde a una regla de dominancia y luego a cada solución se le asigna su valor de aptitud en base a su rango en la población. Algunos típicos esquemas que caracterizan a estos métodos son:

- (a) **Rangos.** Varias propuestas existentes asignan un rango a los individuos respecto a su relación de dominancia con la población. Una primera técnica

³Vector Evaluated Genetic Algorithm.

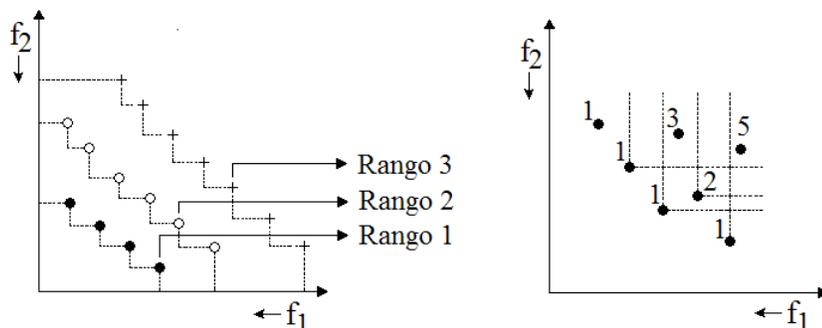


Figura 4.4: Asignación de rangos según Goldberg (izquierda) y según Fonseca y Fleming (derecha).

propuesta por Goldberg [85], consiste en asignar el rango 1 a todos los individuos no dominados de la población. Estos individuos son a continuación suprimidos de la población, y al conjunto siguiente de individuos no dominados les es atribuido el rango 2. El proceso continúa hasta que todos los individuos de la población tengan asignado un rango (figura 4.4 izquierda). Conocidos ejemplos que usan asignación de rangos según Goldberg son NSGA⁴ [185] y NSGAI⁵ [55].

Una segunda técnica propuesta por Fonseca y Fleming [69], estima el rango de un individuo por el número de individuos que dominan al individuo considerado. Por ejemplo, un individuo y_i en la generación t , que está dominado por p_i^t individuos posee un rango dado por $(y_i, t) = 1 + p_i^t$. Todos los individuos no dominados de la población poseen rango 1 (figura 4.4 derecha). Un ejemplo que utiliza este tipo de asignación de rangos es MOGA⁶ [69].

El rango- ϵ -dominancia basado en la ϵ -dominancia de Laumanns et al. [123] (definida en la sección 2.3.3) constituye una tercera técnica que sigue los siguientes pasos: (1) se clasifican todas las soluciones por no-dominancia (figura 4.5 a), (2) se divide todo el espacio de las funciones objetivo en hiper-cajas de tamaño ϵ_j en el j -ésimo objetivo (figura 4.5 b), (3) asumiendo una minimización de todos los objetivos, las hiper-cajas con múltiples soluciones, preservan la solución más cercana a la esquina inferior izquierda de la hiper-caja (figura 4.5 c) y (4) se realiza una ϵ -no-dominancia en

⁴Non-Dominated Sorting Genetic Algorithm.

⁵Elitist Non-Dominated Sorting Algorithm.

⁶Multi Objective Genetic Algorithm.

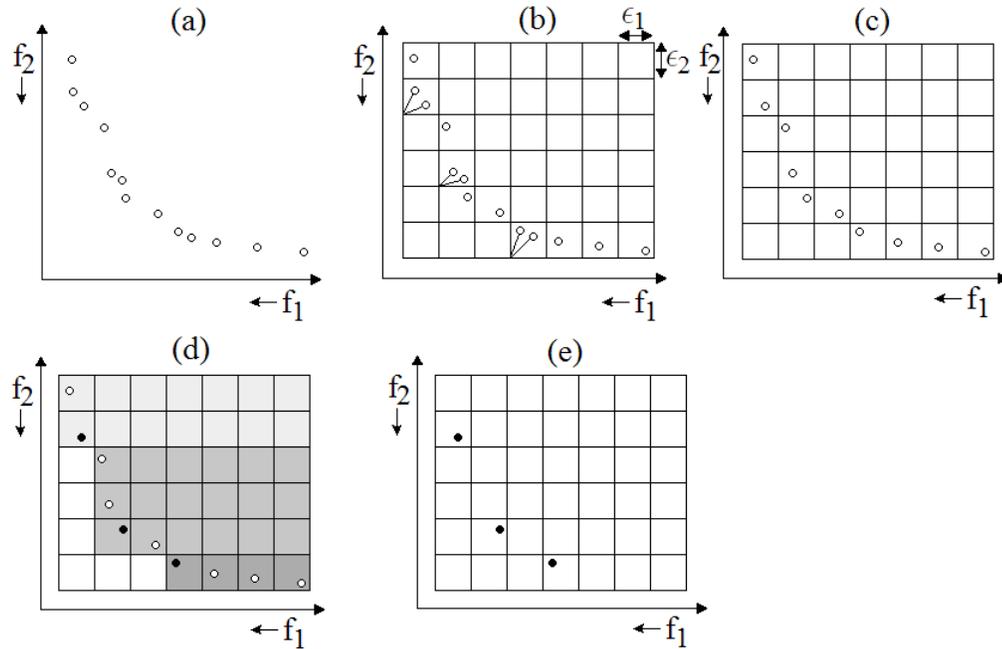


Figura 4.5: Asignación de rangos en base a la ϵ -dominancia de Laumanns.

base a las hiper-cajas (figura 4.5 d y e). Reputados ejemplos que utilizan esta técnica son ϵ -MOEA⁷ [54] y ϵ -NSGAI⁸ [115][116][193].

- (b) **Diversidad.** Los métodos de asignación del valor aptitud basados en rangos presentados anteriormente, tienden a favorecer la convergencia hacia el frente óptimo de Pareto. Sin embargo, estos métodos no garantizan que el frente aproximado obtenido, consiga una buena calidad en términos de una amplia y buena distribución de las soluciones (diversidad) en cualquiera de los espacios de decisión o de los objetivos. Dos de los métodos de mantenimiento de diversidad más representativos en el estado del arte de los AEMOs son el *crowding distance* (distancia de agrupamiento) y el *nearest neighbour* (más próximo vecino).

El *crowding distance* fue introducido por Deb et al. [55] en NSGAI. La distancia de crowding d_i evalúa la densidad de soluciones presentes alrededor de una solución i . La asignación de este valor disminuirá las posibilidades de supervivencia de una solución i presente en una región donde se concentran otras soluciones. A las soluciones extremas se les asigna una distancia

⁷epsilon-Multiobjective Evolutionary Algorithm.

⁸epsilon-Non-Dominated Sorting Genetic Algorithm.

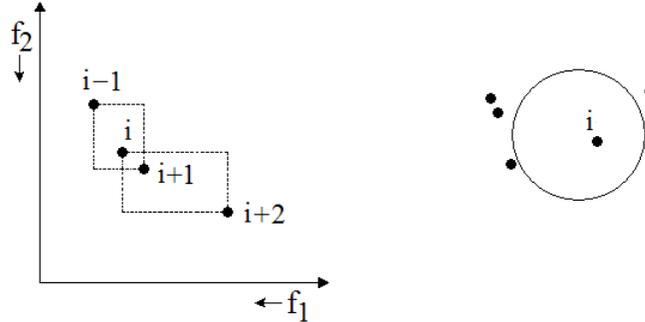


Figura 4.6: La distancia de agrupamiento (crowding distance) utilizado en NSGAI (izquierda) y el más próximo vecino (the nearest neighbour) usado en SPEAI (derecha).

muy grande para darles máxima prioridad. Para una solución intermedia i , la distancia d_i se calcula en función del perímetro del hipercubo de vértices las soluciones más próximas de i sobre cada objetivo. En el ejemplo de la (figura 4.6 izquierda), la solución i será eliminada dado que el perímetro de su cuboide es el más pequeño.

El método del *nearest neighbour*, implementado por Zitzler et al. en SPEAII⁹ [221], mide la densidad en términos de distancias Euclidianas (otras son posibles), de forma que se seleccionan aquellas soluciones con el mayor hiper-esférico volumen vacío alrededor de ellas. En la (figura 4.6 derecha), la solución i será elegida dado su mayor volumen hiper-esférico.

- (c) **Elitismo.** En general, el elitismo es implementado en los AEMOs, utilizando una población externa en la cual se almacenan las soluciones no dominadas encontradas. Esta población es continuamente puesta al día durante el proceso de búsqueda, y soluciones padres almacenadas pueden ser elegidas por el operador selección con el fin de reproducirse y producir soluciones hijos. El uso de elitismo se considera hoy en día imprescindible para la obtención de un buen acercamiento al frente óptimo de Pareto [34] y los algoritmos que lo utilizan obtienen los mejores resultados sobre un gran número de problemas multiobjetivo. Ejemplos de AEMOs elitistas son SPEA [223], SPEAII, PAES¹⁰ [112], NSGAI [55], ϵ -MOEA [54], ϵ -NSGAI [115], Micro Algoritmo Genético [35].

⁹Strength Pareto Evolutionary Algorithm v.2.

¹⁰Pareto Archived Evolution Strategy.

Más recientemente, Coello [34] distingue dos períodos o generaciones en la evolución de los AEMOs y los clasifica según:

1. **AEMOs de primera generación.** Caracterizados por la simplicidad de los algoritmos. Los métodos agregativos, lexicográficos, no Pareto, de Pareto sin elitismo, además de los primeros métodos que introducen preferencias del decisor, formarían parte de esta primera generación.
2. **AEMOs de segunda generación.** Los algoritmos son en este periodo mucho más sofisticados y se caracterizan por la estandarización del mecanismo del elitismo. Coello destaca igualmente en esta segunda generación, la introducción reciente de formas más relajadas de la dominancia de Pareto o ϵ -dominancia [123].

Otra clasificación de los AEMOs [32][99][200], resulta de considerar la toma de decisión final, como resultado de manejar conjuntamente los procesos de optimización y decisión. Ahora, una cuestión fundamental, es en qué momento del proceso toma participación el decisor:

1. **A posteriori.** El decisor manifiesta sus preferencias una vez que el conjunto de soluciones del frente alcanzado le es presentado.
2. **A priori.** El decisor manifiesta sus preferencias, antes del inicio del proceso de búsqueda.
3. **Progresivamente.** El decisor interactúa con el proceso de búsqueda durante su evolución.

Por último, Salazar et al. [171] consideran que en la práctica, cualquier AEMO está embebido en un proceso de toma de decisiones compuesto de tres fases: el Modelo de Toma de Decisión, la Optimización Evolutiva Multiobjetivo (OEM) y la Toma de Decisión (ver la figura 4.7 tomada de [171]).

4.3 AEMOs que asumen preferencias a posteriori

En esta sección, se exponen las originalidades y funcionamiento de algunos de los más destacados AEMOs de primera generación (RWGA [144], VEGA [175], MOGA [69] y NSGA [185]) y de segunda generación (NSGAI [55] y ϵ -MOEA [54]).

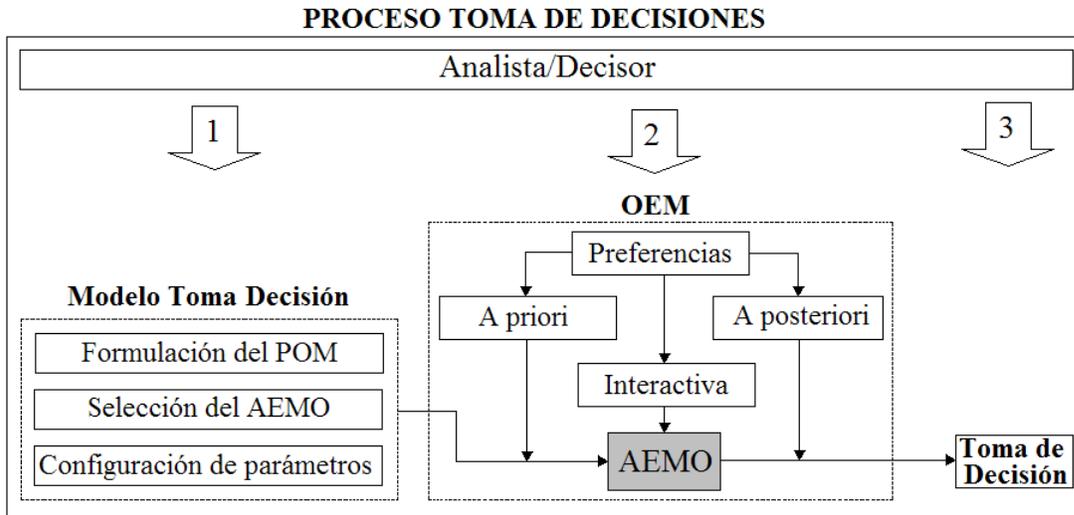


Figura 4.7: Proceso de decisión y AEMOs según Salazar et al.

4.3.1 RWGA

Propuesto en 1995 por Murata e Ishibuchi [144], el método combina las diferentes funciones objetivo del problema, en una única función objetivo (suma ponderada de todas las funciones objetivo). Para que la búsqueda de soluciones no quede fijada en una sola dirección, Murata e Ishibuchi proponen un procedimiento de selección con pesos generados aleatoriamente. Así, cuando un par de individuos son seleccionados para ser cruzados, a cada individuo se le asigna un número aleatorio según la ecuación:

$$w_i = \frac{\text{random}_i}{\sum_{i=1}^n \text{random}_i} \quad i = 1, 2, \dots, n \quad (4.3.1)$$

donde random_i es un número aleatorio entero positivo, n es el número de objetivos y w_i un número real comprendido en el intervalo $[0,1]$. Además, durante la ejecución de RWGA, un conjunto provisional de soluciones Pareto óptimas es almacenado y actualizado en cada generación. De este conjunto, un cierto número de individuos ($N_{\text{élite}}$) son seleccionados aleatoriamente, y usados en cada generación como individuos elitistas. Esta élite de individuos, tiene como efecto preservar la variedad de cada población. El diagrama de flujo del algoritmo se muestra en figura 4.8 y se describe a continuación:

1. **Inicialización:** Se genera un población inicial de individuos de tamaño N_{pop} .

2. **Evaluación:** Se calcula el valor de cada función objetivo para cada individuo, y se actualiza un conjunto provisional de soluciones Pareto óptimas.
3. **Selección:** Se calcula un valor de aptitud para cada individuo, utilizando los pesos aleatorios w_i obtenidos por la ecuación (4.3.1). La probabilidad de selección $P(x)$ de un individuo x en una población ψ está dada por:

$$p(x) = \frac{f(x) - f_{\min}(\psi)}{\sum_{x \in \psi} \{f(x) - f_{\min}(\psi)\}} \quad (4.3.2)$$

donde $f_{\min}(\psi) = \min\{f(x)/x \in \psi\}$. Este paso es repetido para la selección de $\frac{N_{pop}}{2}$ pares de individuos de la actual población.

4. **Cruce:** Para cada par de individuos, se aplica el operador cruce y se generan dos nuevos individuos.
5. **Mutación:** A cada bit de cada individuo generado en el paso anterior, se aplica el operador mutación con una determinada probabilidad.
6. **Estrategia elitista:** Se eliminan aleatoriamente N_{elite} individuos del conjunto N_{pop} generados en los pasos anteriores, y se reemplazan por N_{elite} individuos seleccionados aleatoriamente de la población provisional de soluciones Pareto óptimas actualizada en el paso 2.
7. **Test de terminación:** Si la condición de parada no es satisfecha entonces volver al paso 2.
8. **Decisión:** El AEMO obtiene un conjunto de soluciones Pareto óptimas, a partir de las cuales, un decisor elige la mejor solución acorde a sus preferencias.

Discusión

El método resulta sencillo de implementar. Murata e Ishibuchi aplican el método a un problema de flowshop scheduling, y muestran en su artículo que es capaz de encontrar numerosas soluciones en regiones de Pareto no convexas.

4.3.2 VEGA

VEGA, es en general considerado como el primer algoritmo evolutivo diseñado para la resolución de problemas multiobjetivo. Fue propuesto por Schaffer [174] en 1984. La idea es sencilla y la única diferencia con un algoritmo genético simple es la manera de

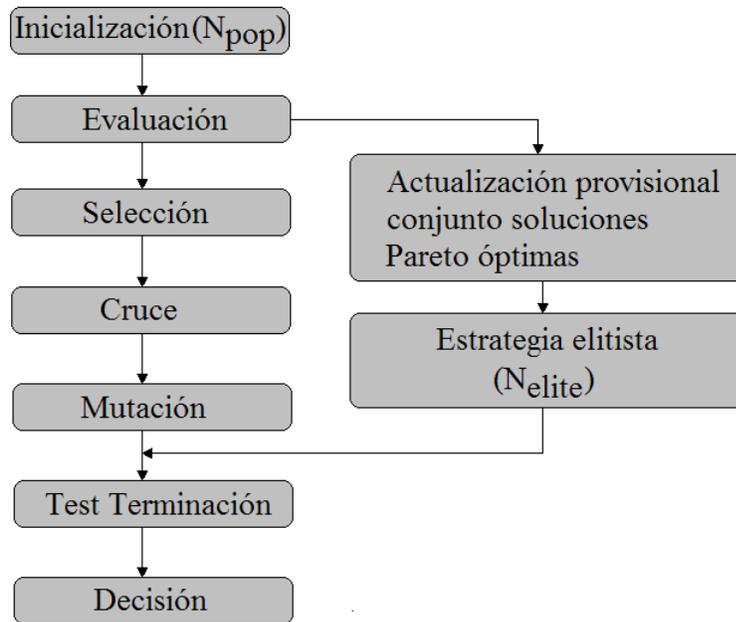


Figura 4.8: Diagrama de flujo de RWGA.

efectuar la selección. VEGA considera una población de N individuos y si se tienen k objetivos, una selección por torneo de $\frac{N}{k}$ individuos es realizada para cada función objetivo. De esta manera, k subpoblaciones son creadas, de forma que cada una de ellas contiene los $\frac{N}{k}$ mejores individuos para una función objetivo k (esta función objetivo determina la aptitud de un individuo dentro de la subpoblación). A continuación las k subpoblaciones son mezcladas, y se termina con la aplicación de los operadores genéticos de cruce y mutación. Las diferentes secuencias del funcionamiento del método se muestran en la figura 4.9.

Discusión

VEGA ha mostrado buenos resultados, sobre todo en problemas donde se buscan buenas soluciones individuales para cada uno de los diferentes objetivos. Sin embargo hay una tendencia a ignorar las soluciones compromiso.

4.3.3 MOGA

En 1993 Fonseca y Fleming [69] propusieron el método MOGA. Este utiliza la noción de dominancia para la evaluación del desempeño de los individuos, basándose en el cálculo de rangos de Fonseca y Fleming tal como se explicó en la sección 4.2. Para determinar la aptitud de un individuo, el método sigue los siguientes pasos:

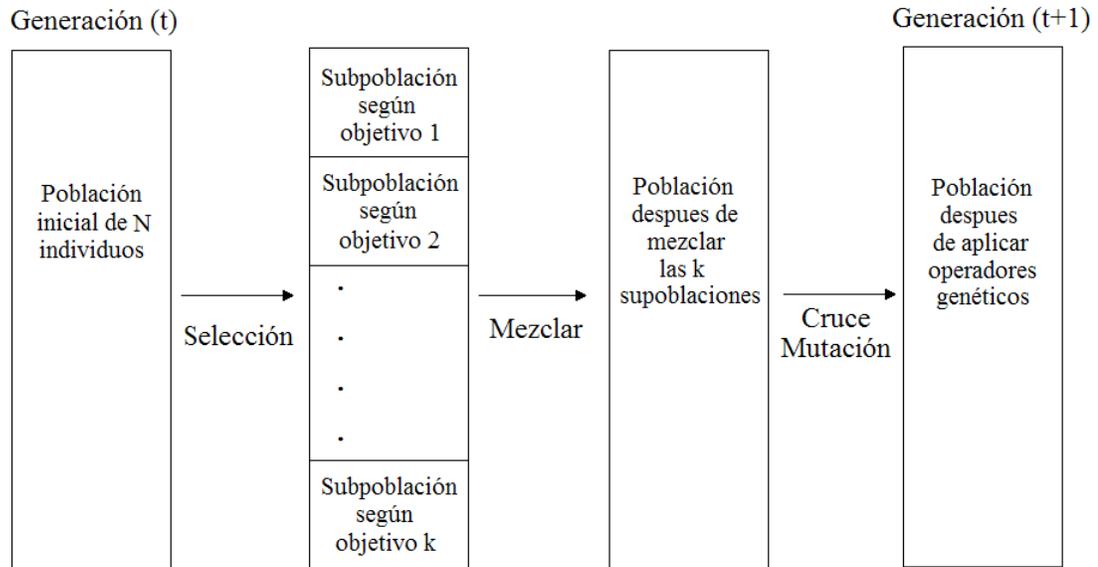


Figura 4.9: Esquema de funcionamiento de VEGA.

1. Calcular el rango de cada individuo de la población.
2. Asignar una aptitud a un individuo, interpolando según una función lineal (aunque otro tipo de función es posible) a partir del mejor (rango 1) hasta el peor (rango $n \leq N$, con $N = \text{número de individuos}$) (ver figura 4.10).
3. Promediar y compartir la aptitud de los individuos con el mismo rango.

Este tipo de asignación de aptitud produce una presión de selección alta, no permitiendo en ciertos casos obtener diversidad en la representación de las soluciones. Para contrarrestar este efecto, Fonseca y Fleming introdujeron una técnica de nichos entre las soluciones del mismo rango. La técnica consiste en calcular los valores de la aptitud compartida $F'(x_i)$ de los individuos con el mismo rango según la Sharing Function Model de Goldberg y Richardson [86] estudiada en la sección 3.7. En cuanto a las distancias de pares de individuos del mismo rango Fonseca y Fleming las estimaron en el espacio de los objetivos según la ecuación de la distancia Euclidiana dada por:

$$d_{ij} = \sqrt{\sum_{K=1}^M \left(\frac{f_k(x_i) - f_k(x_j)}{f_k^{\max} - f_k^{\min}} \right)^2} \quad (4.3.3)$$

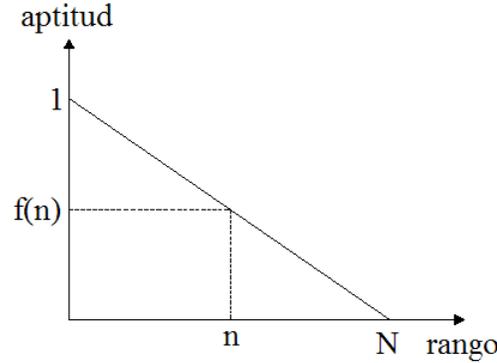


Figura 4.10: Asignación de la aptitud a un individuo en MOGA (se considera una función lineal).

donde M es el número de objetivos, $f_k(x_i)$ y $f_k(x_j)$ son los valores de las funciones objetivo para los individuos x_i y x_j para el objetivo k , y f_k^{\max} y f_k^{\min} son los valores máximos y mínimos de la función objetivo k , alcanzados en la población en curso.

Discusión

MOGA obtiene soluciones de buena calidad y es fácil de implementar, pero al actuar el sharing en el espacio de los objetivos, dos soluciones con idénticos valores en el espacio de los objetivos no pueden estar presentes en la población. Además, los resultados dependen del parámetro σ_{share} utilizado en el sharing.

4.3.4 NSGA - NSGAI

En 1994 Srinivas y Deb [185] realizaron una implementación casi directa del esquema de selección propuesto por Goldberg, que denominaron NSGA. Este método, busca en toda la población los individuos no dominados, los cuales constituyen el primer frente de rango 1. A estos individuos les es asignado un valor de aptitud ficticia que se supone, les da la misma oportunidad de reproducción. Con el fin de mantener la diversidad en la población, se aplica la función de aptitud compartida (ecuación 3.7.1) sobre ese valor de aptitud. Para calcular la Función de Compartición $Sh(d_{i,j})$ (ecuación 3.7.2) se calculan las distancias en el espacio de decisión según la ecuación:

$$d_{ij} = \sqrt{\sum_{K=1}^M \left(\frac{x_k^{(i)} - x_k^{(j)}}{x_k^{\max} - x_k^{\min}} \right)^2} \quad (4.3.4)$$

A continuación, este primer grupo de individuos es suprimido de la población

y se recomienza el procedimiento para determinar los individuos de rango 2. El valor de aptitud atribuido, una vez aplicada la función de aptitud compartida, a este segundo grupo de individuos, es inferior al más pequeño valor aptitud de los del primer grupo. Este mecanismo es repetido hasta que sean tratados todos los individuos de la población.

Una nueva versión de este algoritmo denominado (NSGAI [51][55]) fue presentado en el año 2000 por Deb et al. Se trata de un algoritmo elitista, de manera que en cada nueva generación los mejores individuos encontrados son conservados. Además, el algoritmo integra un operador de selección basado en el *crowding distance* (sección 4.2). Los excelentes resultados obtenidos por NSGAI han hecho que hoy en día sea uno de algoritmos más utilizados en la comunidad AEMO.

La figura 4.11 muestra el funcionamiento del algoritmo. En una generación t , dos poblaciones (P^t y Q^t de tamaños N) coexisten. La población P^t contiene los mejores individuos encontrados hasta la generación t (lo que asegura el elitismo), y la población Q^t está formada por individuos, resultantes de las fases precedentes del algoritmo (selección, cruce y mutación). En primer lugar se crea una población $R^t = P^t + Q^t$, la cual es clasificada según el concepto de rangos de Goldberg (figura 4.4 izquierda), para así identificar los diferentes frentes de soluciones no dominadas (los mejores individuos se encuentran en los primeros frentes). A continuación, se construye una nueva población P^{t+1} , la cual es establecida añadiendo frentes completos (primer frente F_1 de rango 1, segundo frente F_2 de rango 2, etc.) sin sobrepasar el tamaño N . Si el número de individuos presentes en P^{t+1} es inferior a N , el operador *crowding distance* (ilustrado en la figura 4.6 izquierda) es aplicado sobre el siguiente primer frente F_i , no incluido en P^{t+1} . Este operador lo que hace es introducir los $(N - |P^{t+1}|)$ mejores individuos que faltan en la población P^{t+1} . Finalmente, los operadores de selección, cruce y mutación son aplicados sobre los individuos de P^{t+1} , a fin de generar la población de descendientes Q^{t+1} .

Discusión

El método es elitista y no exige ningún parámetro de Sharing. Sin embargo, la aplicación del operador de crowding, puede eliminar soluciones Pareto óptimas y dejar soluciones que si bien son soluciones no-dominadas en la población en curso, no son Pareto óptimas (figura 4.3).

4.3.5 ϵ -MOEA

El método ϵ -MOEA fué introducido por Deb et al. [54] en el año 2003 y está basado en el concepto de ϵ -dominancia de Laumanns et al. [123]. Como se muestra en la figura 4.12, ϵ -MOEA utiliza en una generación t , dos coevolucionarias poblaciones: una población $P(t)$ y una población $E(t)$ o fichero de soluciones ϵ -dominantes. El algoritmo

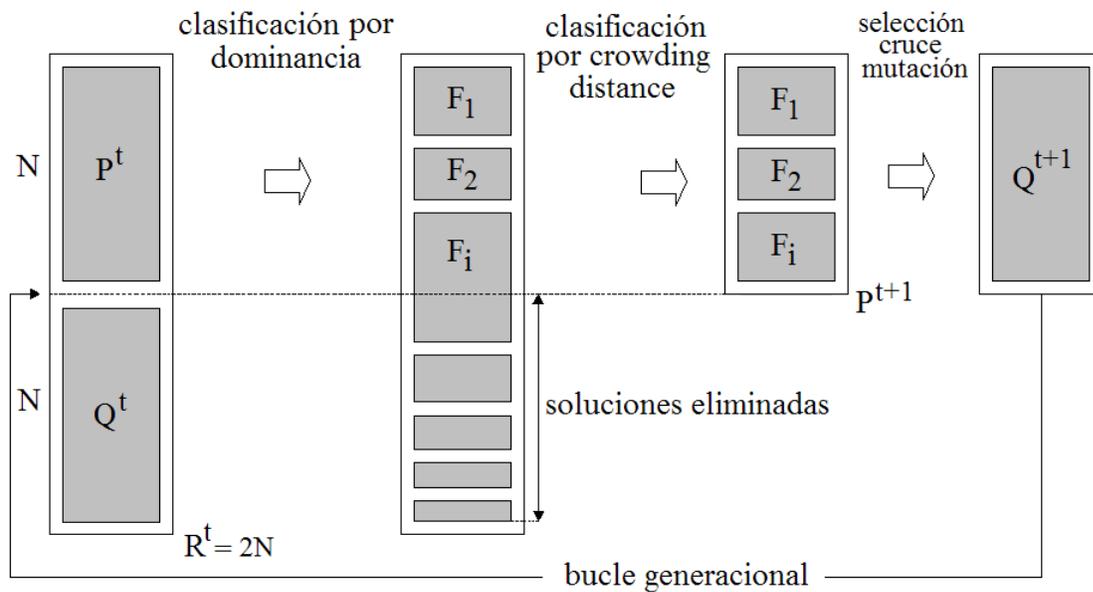


Figura 4.11: Esquema de funcionamiento de NSGAI.

comienza con una población inicial $P(0)$ y un fichero inicial $E(0)$ que contiene las soluciones ϵ -no dominadas de $P(0)$. A continuación dos soluciones, una de la población $P(t)$ y otra del fichero $E(t)$ son seleccionadas para la creación de un descendiente (o varios) h . El proceso de selección de la población, consiste en elegir aleatoriamente dos individuos de $P(t)$ y el concepto de dominancia es aplicado. La solución que domina a la otra es elegida, si ambas son no dominadas se elige al azar a una de ellas. El individuo elegido se denota por p . En el proceso de selección del fichero, se selecciona aleatoriamente un individuo de $E(t)$ denotado e . Después de este procedimiento de selección, se obtiene a partir de p y e una solución descendiente h . Esta es comparada con la población $P(t)$ y el fichero $E(t)$ para su posible inclusión. El procedimiento de inclusión de h en la población consiste en comparar h con todos los miembros de $P(t)$. Si h domina uno o más miembros de la población, entonces h reemplaza a alguno de ellos elegido aleatoriamente. Si algún miembro de la población, domina al descendiente h , entonces este es rechazado. Si ambas posibilidades fallan entonces h sustituye a algún miembro (elegido aleatoriamente) de la población. Para la inclusión de h en el fichero, se compara h con todos los miembros de $E(t)$ utilizando el concepto de ϵ -dominancia. Si h ϵ -domina a algún miembro de $E(t)$ entonces h lo reemplaza, por el contrario h será rechazado si es ϵ -dominado por algún miembro de $E(t)$. Si ninguno de los dos casos ocurre, significa que h comparte la misma hiper-caja con algún miembro de $E(t)$, en cuyo caso ambos son comparados por dominancia. Si h

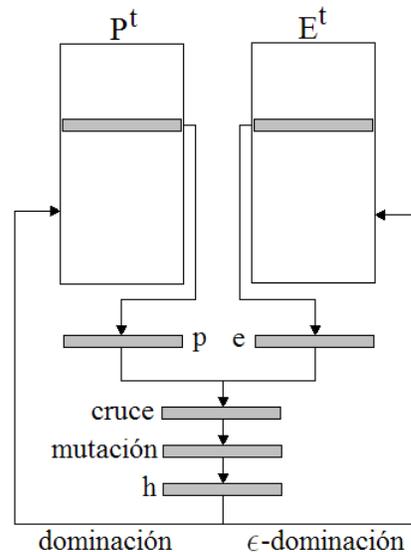


Figura 4.12: Esquema de funcionamiento de ϵ -MOEA.

domina al individuo con el que comparte hiper-caja o h no es dominado por dicho individuo pero se encuentra más cerca de la esquina inferior izquierda (figura 4.5 b y c) de la hiper-caja entonces h es incluido en $E(t)$.

Discusión

El método elige una única solución de cada hiper-caja sobre el frente óptimo de Pareto, manteniendo la diversidad de soluciones sin redundancias en el fichero de soluciones ϵ -dominantes (y por tanto en el frente de Pareto). Eligiendo un apropiado ϵ -vector, se pueden enfatizar unos objetivos más que otros.

4.4 AEMOs que introducen preferencias a priori

Los estudios con AEMOs que articulan preferencias a priori, son hoy en día, bastante menos numerosos que aquellos que reservan las preferencias para una posterior etapa. Sin embargo, recientes trabajos [1][52][53][56], ponen de manifiesto las dificultades en resolver POM con una alta dimensionalidad en el número de objetivos (denominados en inglés *Many Objective Optimisation Problems* [1]). Ahora, procedimientos de búsqueda de una zona de preferencia del frente de Pareto, pueden ser más adecuados. Además Branke [15] [16], señala que si el decisor tiene algún conocimiento del problema, éste debiera ser utilizado durante el proceso de búsqueda, pues de esta forma, el proceso resultaría computacionalmente mucho menos costoso que buscar el frente

completo. Esto, se ha traducido en un notable incremento en los últimos años, tanto en interés (celebración en Honolulu, en abril del 2007: The First IEEE Symposium on Computational Intelligence in Multicriteria Decision Making), como en número de publicaciones que tratan con metodologías evolutivas con un enfoque a priori de las preferencias. Se destacan a continuación las siguientes referencias:

M. Tanaka y T. Tanino [192] en el año 1992, desarrollan (según Coello [34]) el primer esquema que incorpora preferencias en un AEMO.

C.M. Fonseca y P.J. Fleming [69] proponen en el año 1993, modificar el esquema de rangos de Pareto (por ellos propuesto en éste mismo artículo), para incluir metas e información prioritaria acorde a las preferencias de un decisor. El esquema desarrollado, simplifica el proceso de optimización al necesitar el decisor sólo una pequeña porción del frente de Pareto, sin embargo sólo es posible utilizar una única meta, lo que dificulta su utilización en problemas de optimización del mundo real.

E. Sandgren [173] en el año 1994, utiliza programación por metas en un algoritmo genético, para optimizar problemas de estructuras articuladas de barras y mecanismos planos con varios objetivos.

J. Horn [99] en 1997, clasifica los métodos de computación evolutiva considerando que existen tres formas de afrontar los problemas de búsqueda y toma de decisión: (1) decidir antes de buscar, (2) buscar antes de decidir y (3) integrar la búsqueda y la toma de decisión. En su artículo, Horn considera que la manera más común de tratar problemas multicriterio que manejan la toma de decisión antes de la búsqueda, es agregando (agregación escalar o agregación lexicográfica) los múltiples objetivos en un solo objetivo.

D. A. Van Veldhuizen et al. [202] en el año 1997, emplean un algoritmo genético para el diseño geométrico de conexiones de antenas. Utilizan la técnica de la suma ponderada, para combinar los objetivos (ganancia de potencia, simetría, resistencia y reactancia) en una única función objetivo. Los pesos en la función ponderada son utilizados a priori y sus valores son elegidos utilizando el conocimiento adquirido durante las fases preliminares de la investigación.

C.A. Coello [30] en 1998, utiliza un AG para resolver un problema de Optimización Multiobjetivo de diseño de máquinas-herramienta. El método, que es capaz de generar una única solución, combina un vector de pesos para los objetivos (proporcionado por el decisor) con la estrategia Min-Max.

D. A. Van Veldhuizen y G.B. Lamont [200] en el año 1998, realizan una revisión del estado del arte de las técnicas evolutivas multiobjetivo, clasificándolas según: (1) a priori, (2) a posteriori y (3) progresivas.

K. Deb [49] en 1999, utiliza la idea de la programación por metas y modifica el algoritmo NSGA [185] para encontrar un conjunto de soluciones cercanas a la meta o nivel de aspiración del decisor. Tres problemas test y uno de ingeniería (diseño de

una viga soldada) son utilizados para mostrar la eficacia del método propuesto.

En el año 2000, C.A. Coello [32] presenta una revisión del estado del arte de la optimización multiobjetivo evolutiva y el manejo de preferencias, discutiendo algunas de sus ventajas y desventajas.

J. Branke et al. [16] en el año 2001, proponen un algoritmo evolutivo multiobjetivo, capaz de guiar la búsqueda de soluciones hacia una específica región del frente de Pareto. El algoritmo tiene en cuenta las preferencias del decisor, mediante una modificación de la definición de dominancia. Por ejemplo, si se considera un problema con dos objetivos, el decisor tiene que decidir a priori, la máxima cantidad (q_{12}) aceptable de degradación en términos del objetivo 1, la cual puede ser recompensada por una unidad de mejora en términos del objetivo 2 y viceversa (con q_{21} , la máxima cantidad aceptable de degradación en términos del objetivo 2, la cual puede ser recompensada por una unidad de mejora en términos del objetivo 1). Esta información es a continuación utilizada para modificar el esquema de dominancia según:

$$\begin{aligned} x \succ y \Leftrightarrow & (f_1(x) + q_{12}f_2(x) \leq f_1(y) + q_{12}f_2(y)) \\ & \wedge (q_{21}f_1(x) + f_2(x) \leq q_{21}f_1(y) + f_2(y)) \end{aligned} \quad (4.4.1)$$

D. Cvetković y I. Parmee [42] (ver también [39][40][41]) en el año 2002, presentan un método de optimización multiobjetivo ponderado de Pareto, basado en el principio de Pareto y con la posibilidad de especificar la relativa importancia de los objetivos. El método, asigna a cada criterio un peso w_i y redefine la dominancia de Pareto según:

$$x \succeq_w^\tau y \Leftrightarrow \sum_{i=1}^M w_i I \geq (x_i, y_i) \geq \tau$$

donde

$$I \geq (x, y) = \begin{cases} 1, & x \geq y \\ 0, & x < y \end{cases}$$

y τ indica el nivel mínimo de dominancia requerido con $0 < \tau \leq 1$.

R.F. Coelho et al. [28] en el año 2003, aplican a problemas test e industriales, como el diseño de una válvula de purga del vehículo espacial europeo Ariane5, un método multiobjetivo que introduce preferencias del decisor (en base a PROMETHEE [18]), en un algoritmo genético.

K.C. Tan et al. [191] en el año 2003, presentan un algoritmo evolutivo multiobjetivo con un esquema de dominación por metas secuenciales que permite la inclusión de prioridades fuertes/débiles y restricciones en cada uno de los objetivos del problema. Los autores muestran que el método explora, con buenos resultados, la región de interés, con un esfuerzo computacional bajo. Como ejemplo de aplicación utilizan un problema de optimización de diseño de un sistema de servo control para una unidad de disco duro.

K. Deb [50] en el año 2003, propone un AEMO con articulación de preferencias a priori, capaz de focalizar la búsqueda de soluciones en una determinada región (convexa o cóncava) del frente de Pareto. El método, implementado sobre NSGA [185], modifica la función de compartición (3.7.2) (que ahora denomina, función de compartición sesgada o *biased sharing* en inglés), redefiniendo la $d_{i,j}$ (4.3.4) entre dos individuos (que ahora denomina, distancia sesgada o *biased distance* en inglés) en el espacio de los objetivos y ponderándola según:

$$d_{ij} = \sqrt{\sum_{K=1}^M w'_k \frac{(f_k^{(i)} - f_k^{(j)})^2}{(f_k^{\max} - f_k^{\min})^2}}$$

donde $w'_k = \frac{(1-w_k)}{\max_{k=1}^M (1-w_k)}$, son los pesos asignados a cada objetivo por un decisor acorde a sus preferencias. Un problema test y un problema de ingeniería de diseño de una viga soldada, son utilizados para comprobar la eficacia del método.

K. Deb et al. [54] en el año 2003, proponen ϵ -MOEA (expuesto en la sección 4.3.5). Como sugieren los autores, el método permite considerar ϵ_i como una función de f_i , permitiendo centrar la búsqueda en aquella región del frente de Pareto de interés para un decisor.

J. Branke y K. Deb [15] en el año 2004, argumentan que si un decisor tiene alguna idea de qué soluciones pudieran ser preferidas, entonces éste conocimiento debe ser usado, para focalizar la búsqueda en las regiones más interesantes del frente de Pareto. En esta idea, proponen en su artículo:

1. Integrar en NSGAI [51][55], el esquema de dominancia (4.4.1) propuesto en [16].
2. Modificar la distancia de agrupamiento (*crowding distance*) [51][55], por una distancia de agrupamiento sesgada (*biased crowding distance*) dada por la ecuación:

$$D_i = d_i \left(\frac{d'_i}{d_i} \right)^\alpha$$

donde d_i es la original distancia de agrupamiento para la solución i , d'_i la distancia de agrupamiento de la solución i proyectada sobre un plano cuya dirección es especificado por un decisor acorde a la región del frente de Pareto de su interés y α es un parámetro que controla la intensidad del sesgo.

D. Cvetković y C.A. Coello [38] en el año 2005, hacen una revisión de los métodos de toma de decisión multicriterio desde el punto de vista de la escuela Americana (método de la Función de Utilidad) y la escuela Europea (métodos Electre [159][162][164][166][167][215] y Promethee [17][19][20][21]). Además, clasifican las técnicas que integran preferencias en la optimización multiobjetivo evolutiva en: (1) métodos de sumas ponderadas, (2) métodos de Pareto ponderados y (3) métodos coevolucionarios.

Z. Tian y M.J. Zuo [195] en el año 2006, combinan physical programming [138], algoritmos genéticos y niveles de preferencia por cada objetivo (determinados por un decisor), en la optimización multiobjetivo de asignación de redundancia de un sistema multiestado serie-paralelo en el cual hay que maximizar la utilidad y minimizar el coste y peso del sistema simultáneamente.

K. Deb et al. [56] en el año 2006, combinan una estrategia basada en preferencias (en concreto utilizan el enfoque de punto de referencia [209] con las metodologías evolutivas multiobjetivo (en particular usan NSGAI [51][55]), con la idea de encontrar conjuntos de soluciones en las cercanías a puntos de referencia definidos según las preferencias de un decisor. Este método es expuesto en detalle en la sección 4.4.1.

K. Deb y A. Kumar [53] en el año 2007, combinan el método interactivo dirección de referencia de Korhonen y Laakso [117], tomado de la toma de decisiones multicriterio, con las metodologías evolutivas multiobjetivo (en particular utilizan NSGAI [51][55]), para encontrar un conjunto de soluciones en la dirección de referencia. Este método es expuesto en detalle en la sección 4.4.2.

S.F. Adra et al. [1] en el año 2007, remarcan que en general las metodologías evolutivas multiobjetivo tradicionales están focalizadas en problemas con dos o tres objetivos, no pudiendo generalizarlas a problemas reales que requieren la optimización de un número elevado de objetivos. Según S.F. Adra et al., técnicas con articulación de preferencias y en especial técnicas con articulación progresiva de las preferencias del decisor son efectivas. En su artículo, examinan tres recientes trabajos de AEMOs con articulación de preferencias a priori - *Guided Dominance Scheme* de J. Branke [16], *Biased Crowding Technique* de K. Deb [50] y la *Manipulación de la ϵ -Dominancia* de Laumanns [123] en ϵ -MOEA de Deb et al. [54]- y la técnica *Preferability Operator* de C.M. Fonseca y P.J. Fleming [71], considerada como la primera técnica de articulación progresiva de preferencias para optimización evolutiva multiobjetivo.

J. Sanchís et al. [172] en el año 2008, describen una metodología que incorpora preferencias del decisor a priori, physical programming y algoritmos genéticos. Utilizan

para comprobar la eficacia del método, un problema test bi-objetivo y un problema de ingeniería de diseño de una viga, donde se quiere minimizar simultáneamente el área de la viga y su deflexión.

Seguidamente se exponen dos métodos muy recientes (R-NSGAI [56] y RD-NSGAI [53]). Ambos, son AEMOs muy en la línea de los métodos que se presentan en el capítulo 5 como contribuciones de esta tesis.

4.4.1 R-NSGAI

En el año 2006 Deb et al. [56] propusieron un método (R-NSGAI - Reference Point Based NSGAI), el cual introduce preferencias del decisor reflejadas como uno o varios puntos de referencia. El método está basado en el enfoque de punto de referencia sugerido por Wierzbicki [209]. Una interacción t de R-NSGAI transcurre de la siguiente manera: como sucede con NSGAI las poblaciones de padres P^t (de tamaño N) e hijos Q^t son combinadas, y la población resultante $R^t = P^t + Q^t$ (tamaño $2N$) es clasificada según dominancia en frentes. La nueva población P^{t+1} se construye comenzando con los frentes de menor rango hasta llegar a un frente F_i cuyas soluciones no caben completamente en la nueva población. A continuación el operador *preference operator* (operador preferencia) es aplicado al frente F_i con el fin de mantener el tamaño de la nueva población. Finalmente, se genera la población de descendientes Q^{t+1} aplicando los operadores de selección, cruce y mutación a los individuos de P^{t+1} .

El operador *preference operator* sigue los siguientes pasos:

1. Para cada punto de referencia, se calcula su distancia Euclidiana normalizada respecto a cada solución del frente F_i , de forma que para cada punto de referencia, la solución más cercana a dicho punto recibe el valor de *distancia preferida* de 1. De esta forma las soluciones más cercanas a todos los puntos de referencia tienen asignada la más pequeña *distancia preferida*. A continuación, las soluciones con la siguiente distancia más pequeña a cada punto de referencia, reciben el valor *distancia preferida* de 2, y de esta manera el proceso continúa con el resto de las soluciones de F_i . Más tarde, en la generación de la nueva población de descendientes, las soluciones preferidas en la selección por torneo, serán las que tengan un menor valor de *distancia preferida*.
2. Con el fin de mantener diversidad en las soluciones cercanas a cada punto de referencia, la idea de ϵ -cercanía es utilizada. Primero se escoge de forma aleatoria una solución del frente F_i . A continuación, se calcula la distancia Euclidiana en el espacio de los objetivos de todas las soluciones con respecto a la solución escogida. A las soluciones con un valor de distancia igual a ϵ o menos, se les penaliza con un elevado valor de *distancia preferida*, de esta forma sólo una

solución dentro de una ϵ -vecindad, es relevante. El proceso continúa eligiendo al azar, otra solución diferente a la anterior, a la que se aplica de nuevo el concepto de ϵ -cercanía descrito anteriormente.

Discusión

En R-NSGAI, un decisor expresa sus preferencias como uno o varios puntos en el espacio de los objetivos, remarcándose que también es posible enfatizar algunos objetivos más que otros, simplemente reemplazando la distancia Euclidiana por una distancia Euclidiana ponderada. Esta medida de distancia ponderada, puede ser utilizada para encontrar conjuntos de soluciones preferidos en problemas no convexos. Sin embargo el método introduce un nuevo parámetro (ϵ) con el fin de controlar la extensión de las distribuciones de soluciones en las proximidades de los puntos de referencia.

4.4.2 RD-NSGAI

El método RD-NSGAI (Reference Direction Based NSGAI) propuesto por Deb y Kumar en 2007 [53], se inspira en el método interactivo *dirección de referencia para optimización multi-criterio* de Korhonen y Laakso [117], y modifica un AEMO elitista (NSGAI) de la siguiente manera:

1. Un decisor elige dos puntos q_k^0 y q_k^1 en un espacio de k objetivos. A continuación, se calcula el vector dirección de referencia $\vec{s}_k = q_k^1 - q_k^0$.
2. Se escoge un conjunto de puntos ($r(t)$, $t=0,1,\dots$) pertenecientes al vector \vec{s} .
3. La población de individuos $R^t = P^t + Q^t$ (tamaño $2N$) es clasificada en los siguientes términos: para cada punto $r(t)$ en la dirección de referencia, se calcula la distancia Tchebycheff ($d = \frac{\max_{i \in I} |z_i - r_i|}{w_i}$) con respecto a cada solución z de R^t , de forma que las soluciones \bar{z} con la distancia d más pequeña respecto de cada punto $r(t)$, se declaran pertenecer al frente F_1 de rango 1. El siguiente conjunto de soluciones con los valores más pequeños de d , se asignan al frente F_2 de rango 2. El proceso es repetido hasta que todas las soluciones de R^t hayan sido clasificadas.
4. La nueva población P^{t+1} de tamaño N , se establece empezando con las soluciones del frente F_1 , seguidas por las del frente F_2 y así, hasta llegar a aquel frente cuyas soluciones no pueden ser elegidas completamente para mantener el tamaño N de la población P^{t+1} .
5. El operador *crowding distance* es aplicado tal como se hace en NSGAI.

6. Los operadores selección por torneo (son preferidas aquellas soluciones con menor rango, calculadas según paso 3), cruce y mutación son aplicados sobre P^{t+1} para generar la población de descendientes Q^{t+1} .

Discusión

El método puede fácilmente modificarse para considerar múltiples vectores dirección de referencia simultáneamente, de ésta forma múltiples soluciones según cada dirección de referencia podrían ser encontradas. De otra parte, la talla de la población debe ser 2 o 3 veces el número de puntos $r(t)$ considerados en la dirección de referencia, y el costo total del algoritmo puede crecer considerablemente.

4.5 Evaluación de los resultados

Al evaluar los resultados de problemas de optimización multiobjetivo, obtenidos con métodos estocásticos, debe considerarse que varias ejecuciones del método e incluso diferentes métodos, no siempre obtienen los mismos resultados, por tanto, la comparación de la eficacia de estos métodos, hace necesario poder comparar distintos resultados (frente de Pareto).

Un buen número de investigadores han propuesto diferentes métricas. Éstas, comparan y cuantifican la calidad del frente de Pareto obtenido, teniendo en cuenta para ello varios aspectos, fundamentalmente: el acercamiento al frente real, la amplitud del frente y la uniformidad de la distribución de las soluciones.

No existe una métrica universal, pues todas tienen sus ventajas e inconvenientes. Srinivas y Deb [185] utilizaron en 1994, la distribución de desviación Chi-cuadrado, para analizar como se distribuían las soluciones no dominadas a lo largo del frente, obtenidas con NSGA [185]. En la misma línea, Schott [176] propuso en 1995, otra métrica de evaluación de la uniformidad de la distribución de las soluciones del frente obtenido. Un método estadístico basado en el concepto de función de alcance (the attainment function) fue propuesto por Fonseca y Fleming [68] en 1996 (ver también Grunet da Fonseca et al. [93] y Carlos M. Fonseca [67]). El método construye una medida cuantitativa del nivel de alcance de los frentes, por ejemplo se puede intentar determinar alguna región en el espacio de los objetivos con una probabilidad de ser alcanzada tras el 60% de ejecuciones del algoritmo. Hansen y Jaszkiewicz [96] sugirieron en 1998 las R_R métricas para comparar directamente dos conjuntos de soluciones, utilizando funciones de utilidad y así determinar el número esperado de ocasiones en las que las soluciones de un conjunto A serán mejores que las de otro conjunto B. La métrica S (o hipervolumen) fue sugerida por Zitzler y Thiele [222] en 1998. Esta métrica mide en el espacio de los objetivos, el volumen dominado por el frente obtenido. Como la métrica S, la métrica C fue introducida por Zitzler y

Thiele [222] en 1998, se trata de una métrica relativa que compara dos frentes A y B. En 1999 Van Veldhuizen [199] propone varias métricas: error ratio (RE), distancia generacional (generational distance GD), distancia generacional inversa (inverse generational distance IGD), espaciamento (spacing SP) y conjunto generacional de vectores no dominados (overall nondominated vector generation ONVG), las cuales miden diferentes aspectos de la calidad del frente de soluciones obtenido. Zitzler et al. [220] sugirieron en el año 2000 las métricas M_1 , M_2 y M_3 para medir respectivamente la distancia media al frente de Pareto, distribución de las soluciones no dominadas y la amplitud del frente. También en el año 2000 Van Veldhuizen y Lamont [201] propusieron no utilizar una única métrica sino una combinación de ellas y consideraron las métricas GD, SP y ONVG como el conjunto más significativo. Knowles y Corne [113] realizaron en el año 2002 un estudio comparativo de las diferentes métricas utilizando distintos criterios tales como ciclo inducido, medida cardinal, complejidad, etc. Como conclusión recomiendan en su artículo, el uso de las R_1 , R_2 y R_3 métricas de Hansen y Jaszkiewicz [96], y para problemas con un número de objetivos pequeño, la métrica S de Zitzler y Thiele [222]. En 2003 Zitzler et al. [224] compararon los diferentes métodos basándose en los indicadores de calidad unarios¹¹ y en los indicadores de calidad binarios¹². Los autores, además de proponer dos nuevos indicadores (el ϵ -Indicador unario y el ϵ -Indicador binario), muestran las limitaciones de los indicadores de calidad unarios, limitaciones que en principio no poseen los indicadores binarios. Finalizan recomendando el ϵ -Indicador binario. Mas recientemente (en el año 2005), Knowles et al. [114] y Fonseca et al. [72], afirman que básicamente existen dos enfoques en la valoración de las eficacias de los métodos. El primero es el estadístico (the attainment function [68][93][67]). El segundo enfoque, utiliza indicadores de medidas cuantitativas de la eficacia. Los indicadores recomendados son el indicador del hipervolumen o métrica S [222], el ϵ -Indicador unario [224] y los indicadores R_2 y R_3 [96]. A continuación se detallan la métrica C, la métrica S y los ϵ -Indicadores. Las dos primeras han sido utilizadas como medidas de la eficiencia, de los algoritmos propuestos como contribuciones de esta tesis.

4.5.1 La métrica C

Propuesta por Zitzler [222], la métrica C compara dos conjuntos A y B de soluciones no dominadas, de forma que el valor $C(A,B)$ permite calcular mediante la ecuación (4.5.1), el porcentaje de soluciones de B que son débilmente dominadas por al menos

¹¹Resumen la calidad de un conjunto de soluciones no dominadas en un número real.

¹²La calidad del indicador es un número real obtenido en base a comparaciones de pares de conjuntos de soluciones no dominadas.

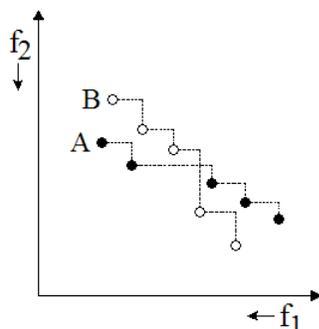


Figura 4.13: Problema en la utilización de la métrica C: $C(A,B)=C(B,A)=\frac{3}{5}$

una solución de A:

$$C(A, B) = \frac{|b \in B / \exists a \in A a \succeq b|}{|B|} \quad (4.5.1)$$

donde $|B|$ representa el número de soluciones contenidas en B.

Cuando $C(A,B) = 1$, todas las soluciones de B son débilmente dominadas por las de A. Si $C(A,B)=0$, ninguna solución de B es débilmente dominada por las de A.

Discusión

Como $C(A,B)$ no es necesariamente igual a $1-C(A,B)$, es preciso calcular siempre $C(A,B)$ y $C(B,A)$. Además, en ciertos casos particulares, el valor $C(A,B)$ no puede diferenciar los dos conjuntos A y B, (figura 4.13).

4.5.2 La métrica S

Sugerida por Zitzler [222], la métrica S o hipervolumen H mide el tamaño de la región dominada por el frente de Pareto y limitada por algún punto de referencia (figura 4.14), Cuando se comparan dos frentes, aquel con el mayor valor H será el preferido. La métrica S, tiene en cuenta [158][199][222] tanto la convergencia hacia el frente de Pareto como la distribución de las soluciones.

Discusión

Una desventaja de esta métrica es su dependencia del punto de referencia. En efecto, cambiando el punto de referencia puede cambiar la conclusión a la que lleva el indicador [72]. Además, el costo computacional es elevado, pues resulta de complejidad exponencial en el número de objetivos y de complejidad polinomial en el número

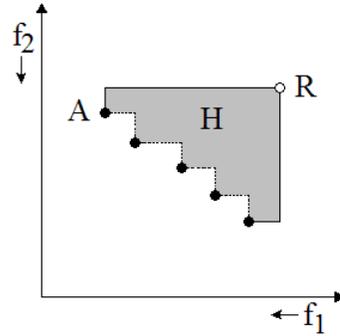


Figura 4.14: Ilustración del hipervolumen H del frente A, limitado por el punto de referencia R.

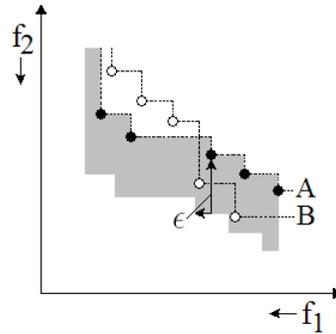


Figura 4.15: Ilustración del ϵ -Indicador binario.

de soluciones del frente [114]. En Fonseca et al. [73] se propone un algoritmo eficaz para su cálculo.

4.5.3 ϵ -Indicadores

ϵ -Indicador binario

Propuesto por Zitzler [224], el ϵ -Indicador binario calcula el valor de ϵ más pequeño que hay que trasladar un conjunto A de soluciones, de forma que toda solución del conjunto B esté cubierta (ver la figura 4.15). Formalmente el ϵ -Indicador binario se define como:

$$I_{\epsilon}(A, B) = \inf_{\epsilon \in \mathbb{R}} \{ \forall z^2 \in B \exists z^1 \in A / z^1 \succeq_{\epsilon} z^2 \} \quad (4.5.2)$$

Discusión

Es necesario calcular $I_\epsilon(A,B)$ y $I_\epsilon(B,A)$. En ciertos casos $I_\epsilon(A,B)=I_\epsilon(B,A)$, no pudiéndose comparar ambos conjuntos A y B.

 ϵ -Indicador unario

Sugerido por Zitzler [224], este indicador está basado en el ϵ -Indicador binario y un frente R de referencia. Formalmente se define como:

$$I_\epsilon(A) = I_\epsilon(A, R) \quad (4.5.3)$$

donde R representa el frente de Pareto o cualquier frente de referencia.

4.6 Problema test

Los problemas test, se han propuesto con el fin de poder comparar y evaluar los diferentes AEMOs existentes. Lo que se pretende es dificultar al método de optimización multiobjetivo, tanto la convergencia hacia el frente óptimo de soluciones, como la diversidad de las soluciones en el frente encontrado.

En 1999, Deb [48] propuso una serie de problemas test bi-objetivos, permitiendo modelar situaciones en las cuales los AEMOs encuentran las dificultades antes mencionadas: converger y asegurar diversidad. Todos ellos están basados en el problema bi-objetivo genérico siguiente:

$$\begin{cases} \min f_1(x) = f(x_1) \\ \min f_2(x) = g(x_2, \dots, x_m)h(f(x_1), g(x_2, \dots, x_m)) \\ \text{con } x = (x_1, \dots, x_m) \end{cases} \quad (4.6.1)$$

donde la función g es la responsable de entorpecer la convergencia mientras la función h dificulta la diversidad. Estos problemas bi-objetivos, fueron posteriormente adaptados por Deb et al. [57] en el año 2001 al caso de k objetivos.

A continuación describimos el problema test ZDT5 de Zitzler, Deb y Thiele [220], que por ser un problema discreto formulado en variables binarias, ha sido utilizado para la validación de las propuestas del capítulo 5 de esta tesis.

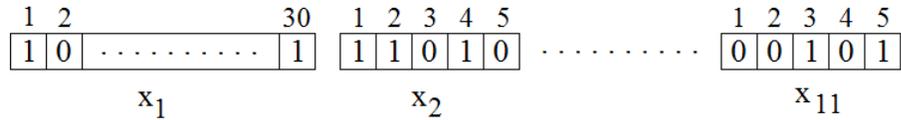


Figura 4.16: Vector solución de ZDT5.

4.6.1 Problema test ZDT5 de Deb

El problema se formula a partir de la ecuación (4.6.1) según:

$$\begin{cases} f(x_1) = 1 + u(x_1) \\ g(x_2, \dots, x_m) = \sum_{i=2}^m v(u(x_i)) \\ h(f_1, g) = \frac{1}{f_1} \end{cases} \quad (4.6.2)$$

donde:

- $u(x_i)$ da el número de unos en la variable binaria x_i .
- $v(u(x_i)) = \begin{cases} 2 + u(x_i) & \text{si } u(x_i) < 5 \\ 1 & \text{si } u(x_i) = 5 \end{cases}$
- el numero de variables es $m=11$.
- $x_1 \in \{0, 1\}^{30}$ y $x_2, \dots, x_m \in \{0, 1\}^5$ (ver la figura 4.16).

El frente real de Pareto es convexo y se obtiene cuando $g(x_2, \dots, x_{11}) = 10$. Los valores de las soluciones se muestran en la tabla 4.1 y su representación gráfica en la figura 4.17.

Tabla 4.1: Soluciones de ZDT5.

f_1	f_2	f_1	f_1
1.0	10.0000	17.0	0.5882
2.0	5.0000	18.0	0.5556
3.0	3.3333	19.0	0.5263
4.0	2.5000	20.0	0.5000
5.0	2.0000	21.0	0.4762
6.0	1.6667	22.0	0.4545
7.0	1.4286	23.0	0.4348
8.0	1.2500	24.0	0.4167
9.0	1.1111	25.0	0.4000
10.0	1.0000	26.0	0.3846
11.0	0.9091	27.0	0.3704
12.0	0.8333	28.0	0.3571
13.0	0.7692	29.0	0.3448
14.0	0.7143	30.0	0.3333
15.0	0.6667	31.0	0.3226
16.0	0.6250		

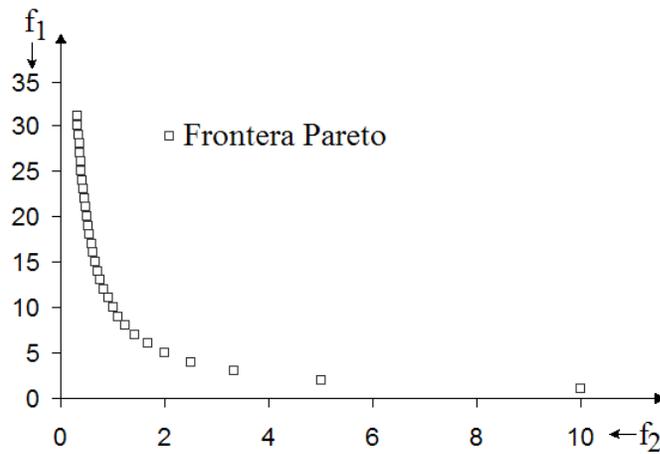


Figura 4.17: Solución gráfica de ZDT5.

Capítulo 5

Contribuciones

5.1 Introducción

En este capítulo se incluyen de forma detallada las contribuciones de esta tesis. Todas las aportaciones, se engloban dentro de las metodologías evolutivas multiobjetivo (en el dominio del discreto) que introducen las preferencias del decisor en una etapa a priori.

La propuesta TOPSISGA¹ (sección 5.2), lidia con las dificultades expuestas en la sección 4.1 y persigue alcanzar una *acentuada densidad* de soluciones en una particular región de la frontera eficiente acorde a las preferencias del decisor. Además, el método aborda la problemática P_γ de Roy (ver la sección 1.6) e induce según el método TOPSIS (sección 1.10.1), un preorden completo en las soluciones alcanzadas.

En la sección 5.3 se propone WP-TOPSISGA², un AEMO orientado a la resolución de problemas industriales, económicos u otros, caracterizados por identificar en los espacios de decisión y objetivos, una solución estable de trabajo o Punto de Funcionamiento (ver la sección 2.3.4). Los propósitos que se persiguen con esta propuesta son: (1) mejorar, si es posible, el Punto de Funcionamiento, (2) obtener una *acentuada densidad* de soluciones en una cierta región (la R_d -PF, ver sección 2.3.4) de la frontera eficiente que dominen al Punto de Funcionamiento y (3) inducir un preorden completo en el conjunto de soluciones logradas.

Remarcamos que los resultados de los algoritmos TOPSISGA y WP-TOPSISGA, han sido publicados como artículos en Méndez et al. [134] y Méndez et al. [135], los cuales se han utilizado en parte, para la elaboración de las secciones 5.2 y 5.3 respectivamente.

¹Multiple-Objective Genetic Algorithm Using the Multiple Criteria Decision Making Method TOPSIS.

²Multi-Objective Genetic Algorithms Using the Working Point and the TOPSIS method.

Dos inquietudes motivan la propuesta CMOEA&TOPSIS³ (sección 5.4): (1) TOPSISGA y WP-TOPSISGA, alcanzan una *acentuada densidad* de soluciones en una porción del frente óptimo de Pareto, aplicando una fuerte presión de selección de individuos en la población (en el transcurso de la búsqueda, se enfatizan mucho más aquellas soluciones ubicadas en la zona de interés que el resto), el interrogante que se plantea es si ¿existe suficiente diversidad en la población para que ésta no se vea ralentizada en su progresión hacia el frente óptimo de Pareto? y (2) un decisor puede indicar qué porción del frente óptimo es de su interés pero no estar seguro. Por esto, la idea que subyace en CMOEA&TOPSIS es encontrar un amplio y visualmente bien distribuido frente de Pareto, pero se hace un especial énfasis, mediante una *acentuada densidad* de soluciones (Soluciones de Precisión SP), en aquella porción del frente (Región de Precisión RP) indicada por el decisor según sus preferencias.

En la sección 5.5 se propone CMOEA20-80&TOPSIS⁴. El algoritmo es en esencia CMOEA&TOPSIS, pero ahora el tamaño SP de la región de precisión se estima automáticamente en cada iteración del algoritmo según la regla 20-80 de Pareto [149].

Para la comparación y evaluación de los métodos expuestos, se han utilizado: el problema test bi-objetivo ZDT5 de Deb et al. [220] expuesto en la sección 4.6, la métrica relativa $C(A,B)$ [222] para medir el porcentaje de soluciones de B que son dominadas por al menos una solución de A (menor valor de $C(A,B)$, mejor) (ver la sección 4.5.1.) y la métrica S (o hipervolumen H) [222] para medir el tamaño de la región dominada por el frente de Pareto y limitada por un punto de referencia (mayor valor de H, mejor) (ver la sección 4.5.2).

5.2 TOPSISGA

5.2.1 Introducción

Muchos de los AEMOs más conocidos y utilizados (por ejemplo NSGAII [51][55]), buscan alcanzar el frente eficiente de soluciones y mostrarlo mediante un conjunto amplio y bien distribuido de soluciones no-dominadas. La idea subyacente es proveer al decisor del más representativo conjunto de soluciones en términos del espacio de los objetivos, reservando la articulación de preferencias para una posterior etapa. Sin embargo, existen situaciones donde el decisor podría estar más interesado en guiar la búsqueda hacia una particular zona de la frontera eficiente. Por ejemplo, en muchos problemas discretos del mundo real, si el conjunto eficiente es muy numeroso y el tamaño de la población no-dominada está limitado, el AEMO no puede contener el

³Combo Multiobjective Evolutionary Algorithm Using the Multiple Criteria Decision Making Method TOPSIS.

⁴Combo 20-80 Multiobjective Evolutionary Algorithm Using the Multiple Criteria Decision Making Method TOPSIS.

conjunto completo de soluciones, obligando a que se pierdan soluciones potenciales de interés para el decisor. Una posible estrategia para superar esta dificultad, es promover aquellas soluciones situadas en la región de interés del decisor mientras que las otras son excluidas durante el proceso de búsqueda. En este sentido, se propone TOPSISGA, un AEMO que mezcla las metodologías evolutivas multiobjetivo (se utiliza NSGAI [51][55], aunque cualquier otra metodología evolutiva multiobjetivo puede ser usada) con el método bien conocido de Toma de Decisiones Multicriterio TOPSIS [101] (sección 1.10.1). El método guía la búsqueda hacia la zona de interés del decisor e induce en las soluciones encontradas un preorden completo, en base a la similaridad [45] al punto ideal (ver el método TOPSIS en la sección 1.10.1).

5.2.2 Algoritmo propuesto

El método propuesto (ver la figura 5.1) tiene una estructura similar a otros AEMOs e introduce dos cambios en la original formulación de NSGAI: primero, el tamaño de la población (Q) y el del fichero de soluciones no dominadas (P) pueden diferir y segundo, el operador *crowding* de NSGAI se ha sustituido por el operador que se ha denominado *similarity* (similaridad). En cada generación t , se forma una población $R^t = P^t + Q^t$ (tamaño N para P^t y tamaño M para Q^t). Como todos los miembros de la población anterior están incluidos en R^t , el elitismo está asegurado. Luego, R^t es clasificada en base a dominancia (figura 4.4 izquierda). La siguiente población P^{t+1} se construye con las soluciones no-dominadas de R^t comenzando con el frente F_1 de rango 1 seguido por el frente F_2 de rango 2 hasta llegar a un frente F_i cuyas soluciones no caben íntegramente en la nueva población (tamaño N). Con el fin de mantener el tamaño N de la nueva población, se aplica al frente F_i el operador *similarity* en los siguientes términos: los extremos del primer frente de rango 1 definen las soluciones ideales y anti-ideales; con esta información se calcula para cada solución, la similaridad [45] a la solución ideal (ver el método TOPSIS en la sección 1.10.1) y a continuación se ordenan las soluciones de mayor a menor según su valor de similaridad, finalmente este criterio es utilizado para elegir las soluciones que entrarán en el frente F_i aunque tienen máxima prioridad las soluciones extremas. Más tarde, se genera una nueva población de descendientes Q^{t+1} aplicando los operadores de selección (torneo binario con el criterio: (1) menor rango, (2) soluciones extremas, (3) mayor valor de similaridad y (4) a igual similaridad se elige al azar), cruce y mutación a los individuos de P^{t+1} .

5.2.3 Resultados experimentales

En esta sección se compara TOPSISGA con NSGAI [51][55]. Se usó un tamaño de población $R=1000$ individuos, una probabilidad de cruce de 0.8, una probabilidad de mutación de 0,01, un valor de la métrica $p=2$ y un número de generaciones

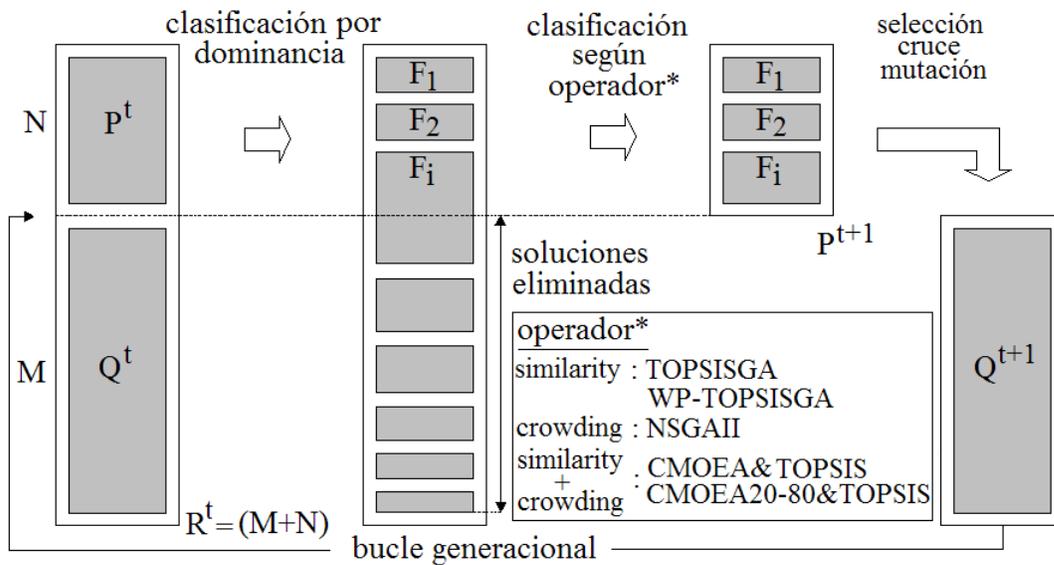


Figura 5.1: Esquema de funcionamiento para TOPSISGA, WP-TOPSISGA, NSGAI, CMOEA&TOPSIS y CMOEA20-80&TOPSIS.

$G=500$. En cuanto al tamaño N de la población no-dominada, se adoptó un valor inferior al tamaño (31) del frente real de Pareto, cambiándose según la secuencia $N=10,15,20,25,30$ individuos.

Resultados visuales de TOPSISGA

Las figuras 5.2, 5.3 y 5.4 muestran los frentes obtenidos con TOPSISGA para diferentes combinaciones de pesos y un tamaño $N=10$ individuos. Se observa cómo el método focaliza la búsqueda de soluciones e induce un preorden en las soluciones (para mejor visualización se muestra el preorden logrado en las cinco primeras soluciones) en la región de interés del decisor; también se percibe cómo una variación de los valores de las preferencias del decisor (pesos), el algoritmo cambia de región a explorar. Por último, la figura 5.5 muestra los frentes alcanzados por TOPSISGA y NSGAI. Mientras TOPSISGA concentra la búsqueda en la región de interés y obtiene un preorden (se representan sólo las cinco primeras soluciones) en las soluciones halladas, NSGAI encuentra un conjunto visualmente bien distribuido.

Comparativa TOPSISGA y NSGAI: métrica C

En la tabla 5.1, se presentan los resultados (media después de diez experimentos) de comparar los frentes obtenidos con TOPSISGA y NSGAI en base a la métrica relativa C . En ella, se observa que el valor $C(A,B)$ o porcentaje de soluciones obtenidas

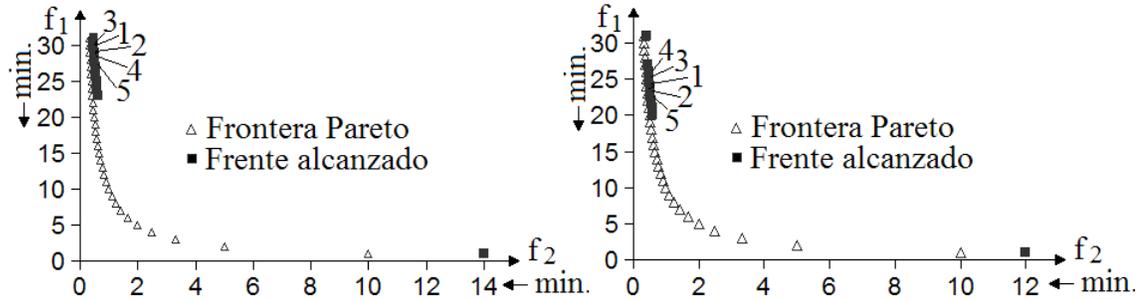


Figura 5.2: Frente óptimo y preorden (según TOPSIS) encontrados con TOPSISGA: $N=10$, $w_1=0.0001$ y $w_2=0.9999$ (izquierda) y $w_1=0.001$ y $w_2=0.999$ (derecha).

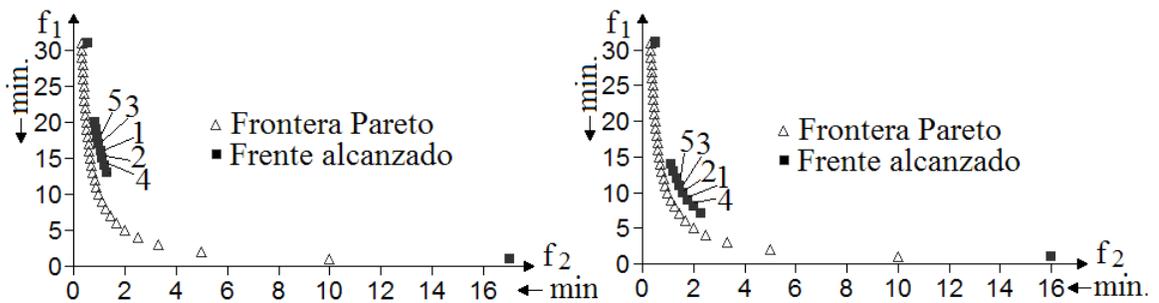


Figura 5.3: Frente óptimo y preorden (según TOPSIS) encontrados con TOPSISGA: $N=10$, $w_1=0.01$ y $w_2=0.99$ (izquierda) y $w_1=0.09$ y $w_2=0.91$ (derecha).

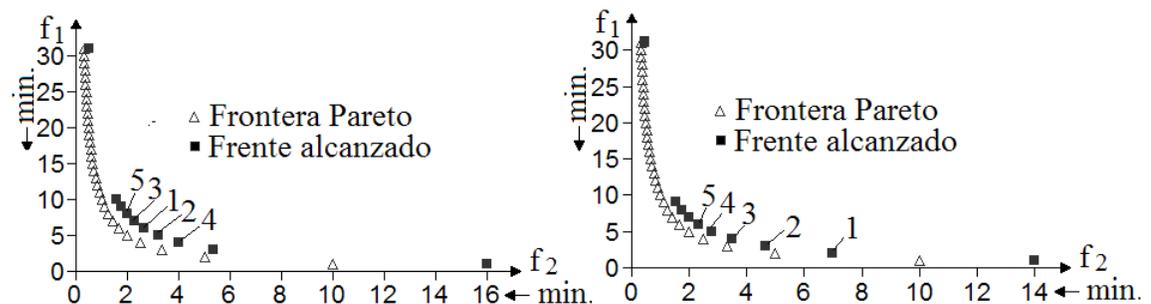


Figura 5.4: Frente óptimo y preorden (según TOPSIS) encontrados con TOPSISGA: $N=10$, $w_1=0.5$ y $w_2=0.5$ (izquierda) y $w_1=0.99$ y $w_2=0.01$ (derecha).

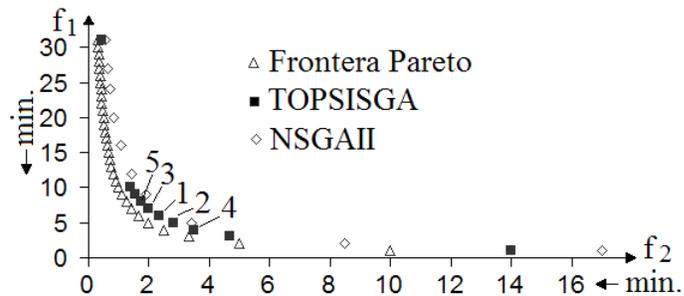


Figura 5.5: Comparación de los frentes alcanzados por TOPSISGA y NSGAI con $N=10$ individuos. Además, para TOPSISGA se muestra el prorden (según TOPSIS) encontrado en las soluciones cuando los valores de los pesos son $w_1=w_2=0.5$.

Tabla 5.1: Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes alcanzados con TOPSISGA (frente B) y NSGAI (frente A).

	N=10	N=15	N=20	N=25	N=30
$C(A,B)$	8	29.32	49	40	40
$C(B,A)$	26	13.34	30	39.2	40.8

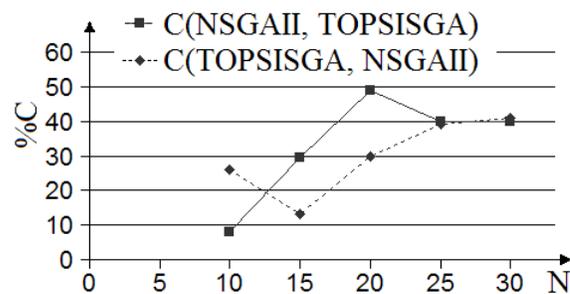


Figura 5.6: Visualización gráfica de los resultados de la tabla 5.1.

con TOPSISGA (frente B) dominadas por las soluciones alcanzadas con NSGAI (frente A) es menor para valores $N=10$ individuos de la población no-dominada. Sin embargo, el valor $C(B,A)$ o porcentaje de soluciones obtenidas con NSGAI (frente A) dominadas por las soluciones alcanzadas con TOPSISGA (frente B) es menor para valores de $N=15,20$ individuos. Finalmente, cuando el valor N es próximo al tamaño (31 individuos) de la frontera real del problema, los valores $C(A,B)$ y $C(B,A)$ son parecidos.

Para una mejor interpretación de los valores de la tabla 5.1, en la figura 5.6 se muestran gráficamente los resultados de dicha tabla.

5.2.4 Conclusiones

Se propone TOPSISGA, un AEMO que combina la segunda generación de AEMOs (en particular se utiliza NSGAI, aunque cualquier otra metodología evolutiva multiobjetivo puede ser usada) con el método de la toma de decisiones multicriterio TOPSIS. Los experimentos muestran que: (1) el porcentaje $C(A,B)$ (media después de diez tiradas) de soluciones obtenidas con TOPSISGA (frente B) que son dominadas por las obtenidas con NSGAI (frente A), es menor para valores $N=10$ individuos de la población no-dominada, (2) el porcentaje $C(B,A)$ (media después de diez tiradas) de soluciones obtenidas con NSGAI (frente A) que son dominadas por las obtenidas con TOPSISGA (frente B), es menor para valores de $N=15,20$ individuos y (3) cuando el valor N de la población no-dominada es próximo al tamaño de la frontera real del problema, los valores $C(A,B)$ y $C(B,A)$ son similares (remarcamos que para tamaños N de la población no-dominada iguales o superiores al tamaño real del frente de Pareto, los operadores similarity de TOPSISGA o crowding distance de NSGAI están muy limitados, de ahí la similitud de los resultados). Además, TOPSISGA concentra la búsqueda en la región de interés, dando un preorden a la lista de soluciones en base a la similaridad [45] a la solución ideal y acorde con las preferencias del decisor. No obstante, es conveniente hacer balance entre el amplio frente producido por NSGAI con la identificación y exploración de la zona de interés realizada por TOPSISGA.

5.3 WP-TOPSISGA

5.3.1 Introducción

Existen problemas industriales, económicos u otros, caracterizados en el espacio de los objetivos por un Punto de Funcionamiento (PF) [134][136] (sección 2.3.4), el cual identifica una solución estable y realizable de trabajo no necesariamente eficaz o Pareto óptima. En este tipo de problemas, la utilización de AEMOs, caracterizados por su aptitud en encontrar amplios y bien distribuidos frentes de soluciones como

paso previo al proceso de toma de decisiones, puede no ser lo más adecuado. De hecho, un decisor inmerso en la tarea de mejorar el PF, puede estar más interesado en un pequeño número de soluciones, por ejemplo, aquellas que localizadas en una cierta región del frente óptimo (la R_d -PF [134], ver sección 2.3.4) dominan el PF. En este sentido, se propone WP-TOPSISGA, un algoritmo que combina la segunda generación de AEMOs (usamos NSGA-II, aunque otro método evolutivo multiobjetivo de segunda generación puede ser utilizado), el método de Toma de Decisiones Multicriterio TOPSIS [101] y el PF. La idea que subyace, es conducir la búsqueda de soluciones hacia la región de interés (la R_d -PF) y clasificar las soluciones halladas según el método TOPSIS. Es importante notar, que con el método TOPSIS se necesitan conocer las preferencias o pesos otorgados por un decisor a los objetivos del problema, sin embargo en el método aquí propuesto, los pesos se estiman de forma automática por interpolación en cada iteración del algoritmo en base al valor del PF.

5.3.2 Algoritmo propuesto

El algoritmo WP-TOPSISGA (figura 5.1) propuesto, difiere del de TOPSISGA (sección 5.2.2) en cómo se obtienen los valores de los pesos w_i para el cálculo de las distancias TOPSIS ponderadas (ver método TOPSIS en la sección 1.10.1) necesarias para el computo de la similaridad de cada solución. Con TOPSISGA los valores de los pesos son estimados por el decisor acorde a sus preferencias. Con WP-TOPSISGA los valores w_i , son estimados de forma automática por interpolación en cada iteración del algoritmo en base al valor del PF, de la siguiente manera: si suponemos una cierta generación o instante t de la búsqueda y un frente óptimo resultante, como el que se muestra en la figura 5.7 (en términos de maximización), la elección por un decisor de la solución extrema A, indicaría que el objetivo f_2 le resulta de máxima relevancia comparada con el objetivo f_1 (en valores cuantitativos de las preferencias: $w_1=0$, $w_2=1$); justo lo contrario si la elección del decisor fuese la solución extrema F, ahora es f_1 comparado con f_2 , el objetivo de máxima relevancia (en valores cuantitativos de las preferencias: $w_1=1$, $w_2=0$). Ambas soluciones extremas A y F, definen los extremos de un intervalo cuyos valores son conocidos, si ahora la elección del decisor fuese la solución D (solución más cercana al PF), el valor de w_2^D puede ser estimado por interpolación (se utilizó, el método de los coeficientes indeterminados) según la

ecuación $w_2^D = \frac{f_2^D - f_2^F}{f_2^A - f_2^F}$ y el valor de w_1^D mediante $w_1^D = 1 - w_2^D$. Además, el operador similarity de TOPSISGA se ha modificado de forma que las soluciones que formarán parte de frente F_i (para mantener el tamaño N de la población) siguen este orden: (1) las soluciones extremas (2) soluciones de mayor valor de similaridad y que pertenecen a la R_d -PF y (3) resto de soluciones con mayor valor de similaridad. En cuanto al

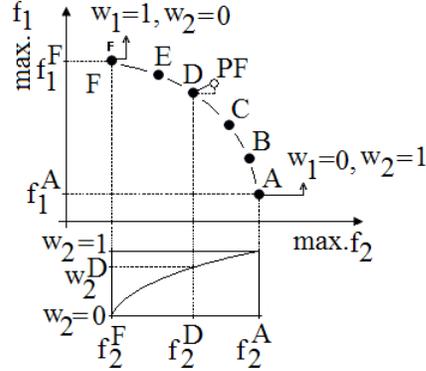


Figura 5.7: Estimación de los pesos w_i según WP-TOPSISGA.

operador de selección, se utilizó torneo binario con el criterio: (1) individuo de menor rango, (2) soluciones extremas, (3a) si los individuos pertenecen a la R_d -PF se elige el de mayor valor de similaridad o al azar, si ambos individuos tienen el mismo valor de similaridad, (3b) si los individuos no pertenecen a la R_d -PF se elige el de mayor valor de similaridad o al azar, si ambos individuos tienen el mismo valor de similaridad y (3c) si un individuo pertenece a la R_d -PF y el otro no, se elige el que pertenece a la R_d -PF.

5.3.3 Resultados experimentales

WP-TOPSISGA se compara en ésta sección con R-NSGAI [56]. Para ambos algoritmos el tamaño de la población fue fijado en $R=1000$ individuos y el del fichero de soluciones no-dominadas en $N=10,15,20,30$ individuos, las probabilidades de cruce y mutación se establecieron en los valores 0.8 y 0.01 respectivamente, el valor de la métrica p se fijó en el valor 2 y el del número máximo de generaciones en $G=500$. Como soluciones del PF se eligieron los valores $(f_1=25, f_2=0.8889)$, $(f_1=16, f_2=0.9286)$, $(f_1=16$ y $f_2=1.0714)$ y $(f_1=5$ y $f_2=2.6)$. Para R-NSGAI, el valor ϵ -cercanía se estableció en 0,0001.

Resultados visuales: WP-TOPSISGA - R-NSGAI

Las figuras 5.8 ($N=10$ individuos, $f_1=25$ y $f_2=0.8889$), 5.9 ($N=15$ individuos, $f_1=25$ y $f_2=0.8889$), 5.10 ($N=20$ individuos, $f_1=25$ y $f_2=0.8889$), 5.11 ($N=10$ individuos, $f_1=16$ y $f_2=0.9286$), 5.12 ($N=15$ individuos, $f_1=16$ y $f_2=0.9286$), 5.13 ($N=20$ individuos, $f_1=16$ y $f_2=1.0714$), 5.14 ($N=10$ individuos, $f_1=5$ y $f_2=2.6$), 5.15 ($N=15$ individuos, $f_1=5$ y $f_2=2.6$) y 5.16 ($N=20$ individuos, $f_1=5$ y $f_2=2.6$) muestran los frentes óptimos alcanzados por WP-TOPSISGA y R-NSGAI. Remárguese como WP-TOPSISGA

Tabla 5.2: Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes alcanzados con WP-TOPSISGA (frente B) y R-NSGAI (frente A).

	$f_1=25, f_2=0.8889$	$f_1=16, f_2=0.9286$	$f_1=16, f_2=1.0714$	$f_1=5, f_2=2.6$
C(A,B) (N=10)	0	0	40	0
C(B,A) (N=10)	50	0	40	0
C(A,B) (N=15)	0	0	40	0
C(B,A) (N=15)	48.82	0	40	0
C(A,B) (N=20)	0	40	40	40
C(B,A) (N=20)	30	40	40	40
C(A,B) (N=30)	0	0	0	0
C(B,A) (N=30)	0	0	0	0

focaliza mucho mejor las soluciones en la región de interés (la R_d -PF) especialmente cuando el PF se sitúa hacia el extremo superior del frente. Nótese también el preorden inducido en las soluciones con WP-TOPSISGA.

Comparativa WP-TOPSISGA y R-NSGAI: métrica C

Los resultados (media después de diez experimentos) de comparar WP-TOPSISGA y R-NSGAI en base a la métrica relativa C , se muestran en la tabla 5.2. Se observa que para tamaños pequeños ($N=10,15,20$) de la población no-dominada, el valor $C(A,B)$ o porcentaje de soluciones obtenidas con WP-TOPSISGA (frente B) dominadas por las soluciones alcanzadas con R-NSGAI (frente A) es menor (cuanto menos mejor) para valores del PF escorados hacia el extremo superior del frente de Pareto (punto $f_1=25, f_2=0.8889$). Sin embargo, los valores $C(A,B)$ y $C(B,A)$ son similares al alejarse el PF del extremo superior del frente (puntos $f_1=16, f_2=0.9286$, $f_1=16, f_2=1.0714$ y $f_1=5, f_2=2.6$) y tamaños N de la población no-dominada, cercanos al tamaño (31) de la frontera real del problema.

5.3.4 Conclusiones

Muchos problemas de ingeniería, económicos u otros, identifican una solución (Punto de Funcionamiento PF) estable y realizable en el espacio de los objetivos, la cual no es necesariamente eficaz o Pareto óptima. En estos problemas, el uso de AEMOs (caracterizados por encontrar amplios y bien distribuidos frentes de soluciones, como paso previo a la toma de decisión) puede no ser lo más apropiado. Por ejemplo, un decisor que busca mejorar el PF, puede estar más interesado en un pequeño conjunto de soluciones, por ejemplo en aquellas que dominan al PF (soluciones que pertenecen a

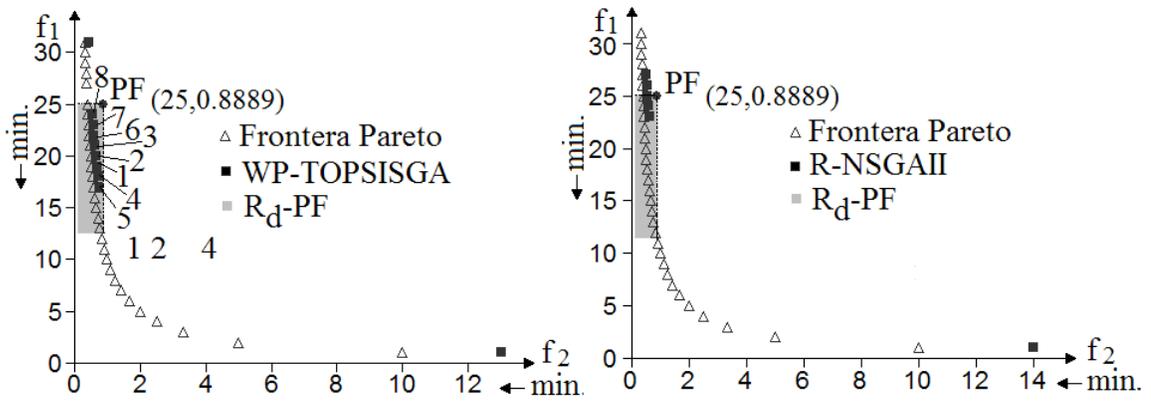


Figura 5.8: Frente óptimo (N=10 individuos) según WP-TOPSISGA y R-NSGAI.

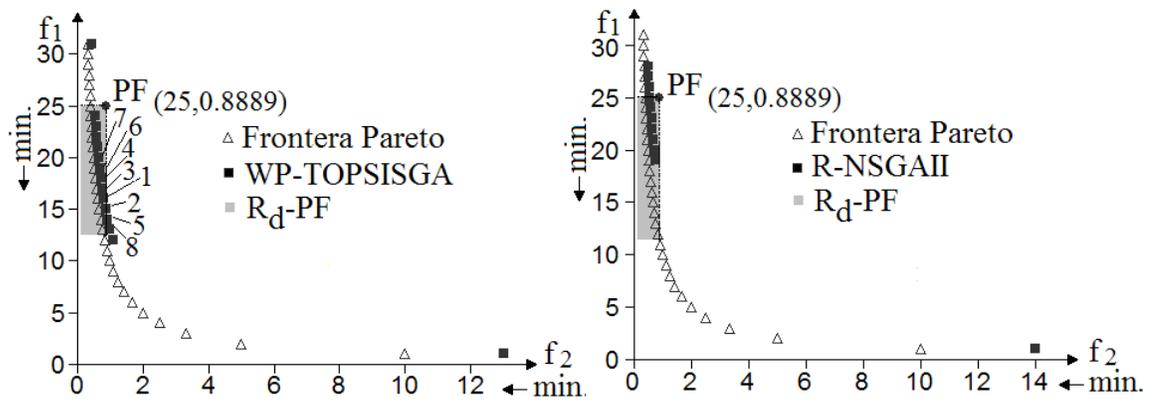


Figura 5.9: Frente óptimo (N=15 individuos) según WP-TOPSISGA y R-NSGAI.

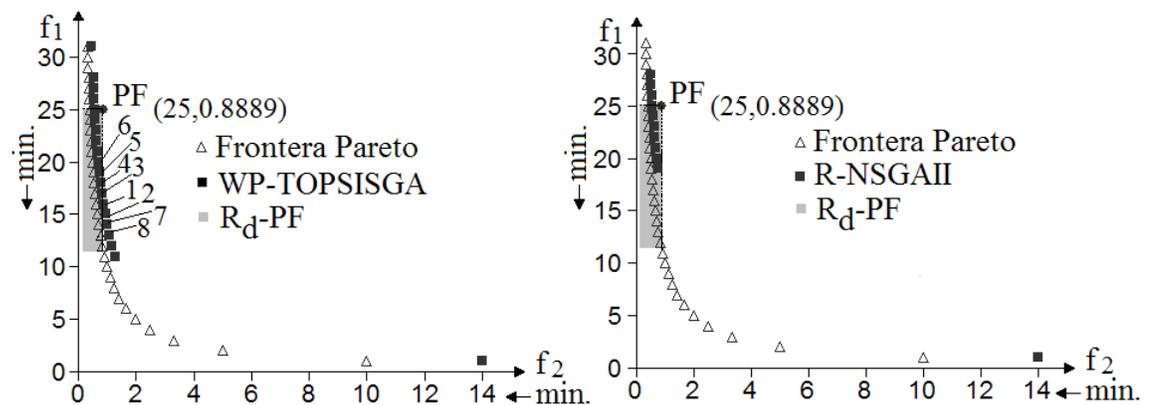


Figura 5.10: Frente óptimo (N=20 individuos) según WP-TOPSISGA y R-NSGAI.

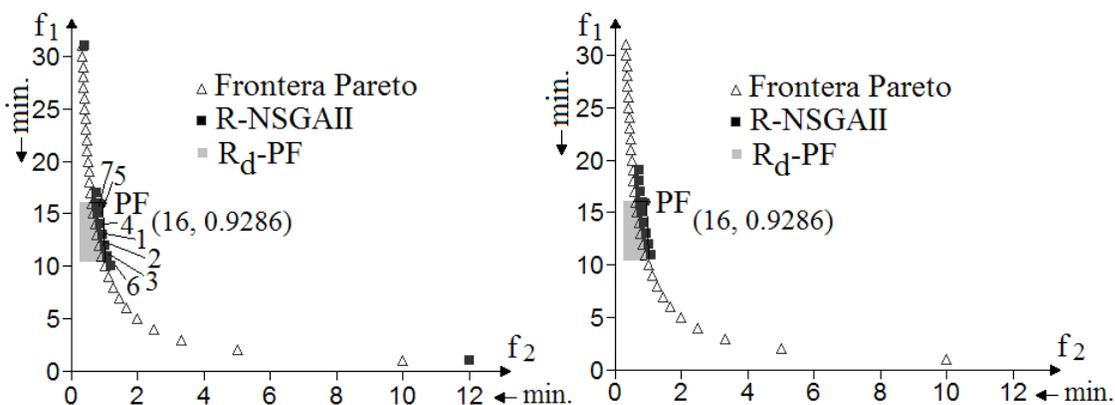


Figura 5.11: Frente óptimo ($N=10$ individuos) según WP-TOPSISGA y R-NSGAI.

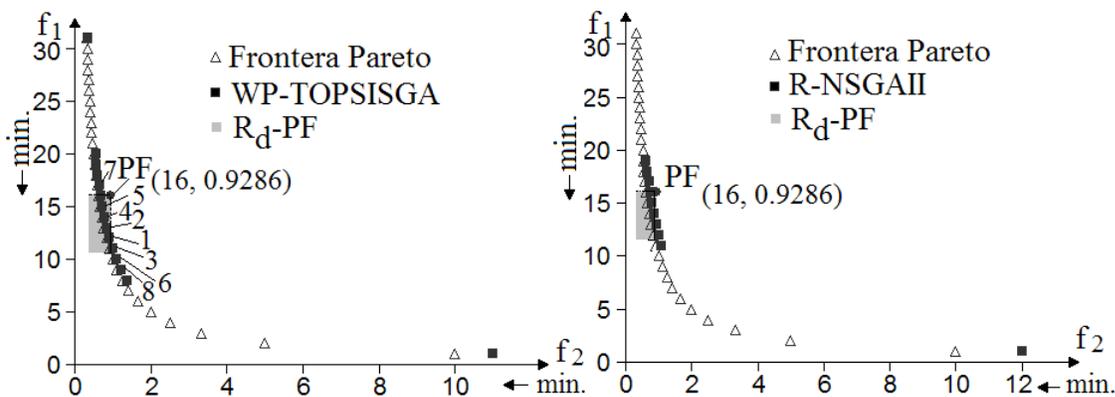


Figura 5.12: Frente óptimo ($N=15$ individuos) según WP-TOPSISGA y R-NSGAI.

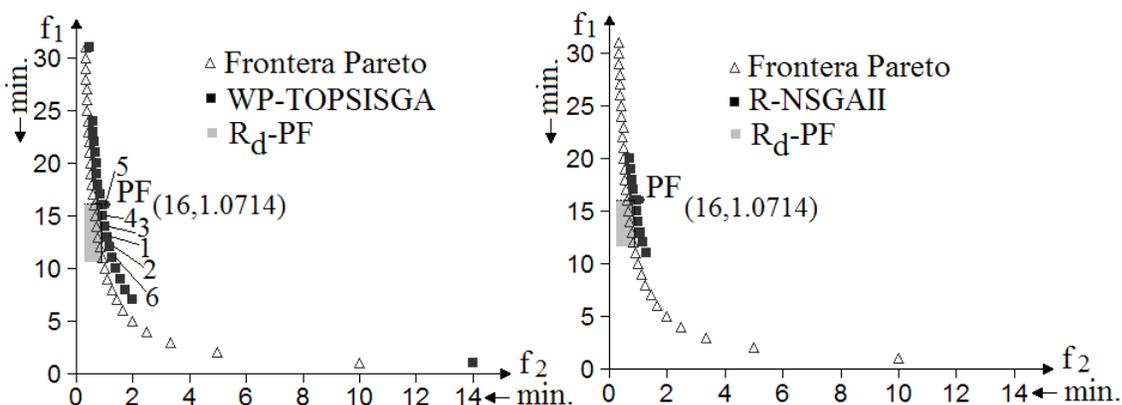


Figura 5.13: Frente óptimo ($N=20$ individuos) según WP-TOPSISGA y R-NSGAI.

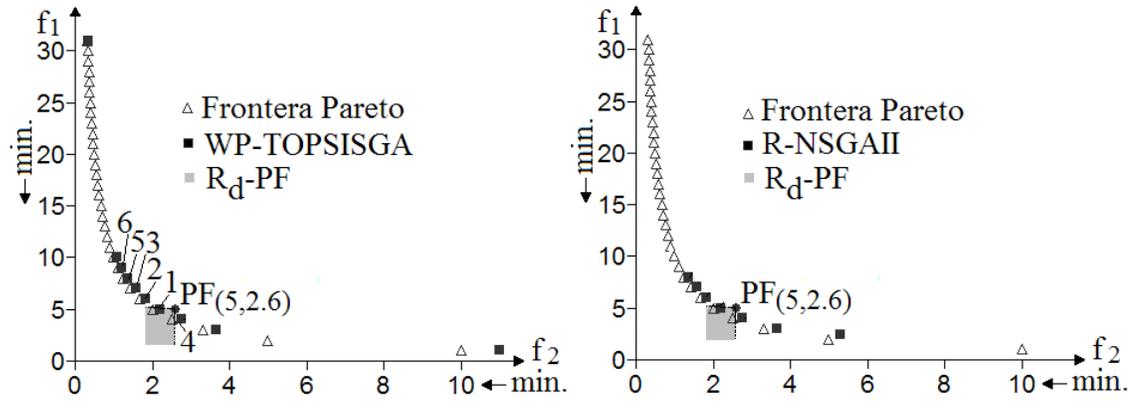


Figura 5.14: Frente óptimo (N=10 individuos) según WP-TOPSISGA y R-NSGAI.

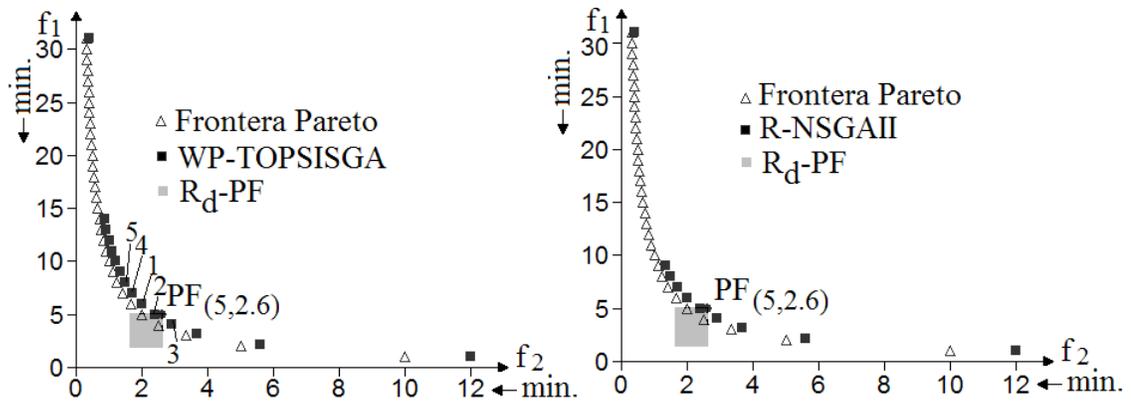


Figura 5.15: Frente óptimo (N=15 individuos) según WP-TOPSISGA y R-NSGAI.

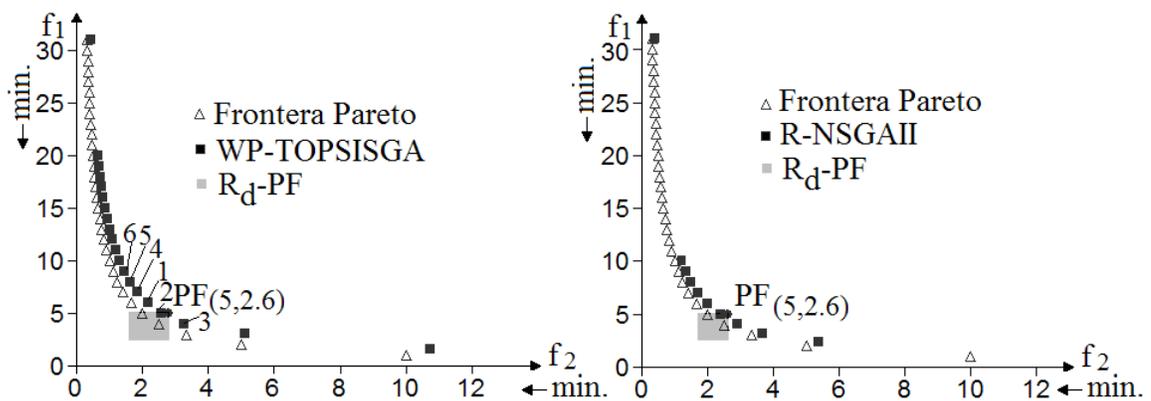


Figura 5.16: Frente óptimo (N=20 individuos) según WP-TOPSISGA y R-NSGAI.

la región dominante del Punto de Funcionamiento o R_d -PF). En esta idea, se propone WP-TOPSISGA, un algoritmo que mezcla las técnicas de los AEMOs de segunda generación, el método TOPSIS (un método de estimación de los pesos por interpolación es aquí aplicado) y el PF. Los experimentos revelan que WP-TOPSISGA concentra las soluciones encontradas en la R_d -PF y las clasifica en términos de la similaridad [45] a la solución ideal. Además, el método fue comparado con R-NSGAI en base a la métrica $C(A,B)$ donde B representa el frente alcanzado con WP-TOPSISGA y A el alcanzado con R-NSGAI. Para los casos considerados, los valores de la métrica C, son mejores para WP-TOPSISGA que para R-NSGAI cuando se eligen valores del PF escorados hacia el extremo superior del frente de Pareto y tamaños de la población no dominada, alejados del tamaño del frente óptimo real ($N=10,15,20$ individuos). Sin embargo, al alejar el PF del extremo superior del frente de Pareto, los valores de la métrica C son similares para ambos algoritmos. Remarcamos que al igual como sucede con TOPSISGA (sección 5.2), para tamaños N de la población no-dominada iguales o superiores al tamaño real (31 soluciones) del frente de Pareto, los operadores similarity de WP-TOPSISGA o preference operator de R-NSGAI están fuertemente limitados, lo que explica que los resultados de la métrica C sean iguales con $N=30$ individuos en la tabla 5.2.

5.4 CMOEA&TOPSIS

5.4.1 Introducción

Dos objetivos fundamentales se persiguen con CMOEA&TOPSIS. En primer lugar, se pretende encontrar un mejor equilibrio exploración-explotación que el logrado con TOPSISGA (este método busca una acentuada densidad de soluciones en una zona específica del frente óptimo, pero en contrapartida, aplica una fuerte presión de selección enfatizando mucho más aquellas soluciones de la zona de interés que al resto) y en segundo lugar, se afronta aquella situación en la que un decisor indica que parte de frente óptimo es de su interés, pero no está seguro. La estrategia adoptada ha sido la de juntar en un único algoritmo (algoritmo combo) las prestaciones de los mejores AEMOs a posteriori de segunda generación (hemos utilizado NSGAI, si bien otro método puede ser adoptado) y las de los AEMOs a priori (hemos utilizado en este caso TOPSISGA, aunque otro método de toma de decisiones se puede utilizar). El método encuentra un amplio y bien distribuido frente óptimo de soluciones, en el cual se distinguen dos zonas: la primera (la Región de Precisión RP), de tamaño el número de Soluciones de Precisión (SP) elegido por el decisor, se caracteriza por su acentuada densidad de soluciones (el operador *similarity* de TOPSISGA es aplicado), por el contrario, la segunda identifica un frente diverso y bien distribuido (el operador

crowding distance de NSGAI es ahora aplicado). Finalmente, CMOEA&TOPSIS aborda la problemática P_γ de Roy y clasifica las soluciones de la RP según TOPSIS [101].

5.4.2 Algoritmo propuesto

CMOEA&TOPSIS sigue los pasos de TOPSISGA y NSGAI (figura 5.1), pero cuando las soluciones de un frente F_i no caben al completo en la nueva población P^{t+1} , se aplican para mantener el tamaño N (se matiza que las soluciones extremas tienen preferencia) los operadores *similarity* y *crowding distance* en los siguientes términos: el frente F_i es clasificado por similaridad a la solución ideal y las soluciones con mayor valor de similaridad pasan a formar parte de la nueva población hasta un máximo de valor SP definido por el decisor, si la población continúa sin completarse, se eligen del resto de soluciones del frente F_i aquellas que resultan ser las mejores en la clasificación por *crowding distance*. Más tarde, en la selección de soluciones por torneo para la generación de la nueva población de descendientes, las soluciones preferidas se escogen según: (1) menor valor de rango, (2) soluciones extremas, (3a) mayor valor de similaridad (si la selección se efectúa entre soluciones que han sido clasificadas por similaridad), (3b) al azar (si la selección se efectúa entre soluciones que han sido clasificadas por distancia *crowding*) y (3c) solución clasificada por similaridad (si la selección se efectúa entre una solución clasificada por similaridad y otra clasificada por distancia *crowding*).

5.4.3 Resultados experimentales

Se exponen aquí los resultados de comparar CMOEA&TOPSIS con los algoritmos TOPSISGA (sección 5.2) y NSGAI [51][55]. En todos los casos se utilizó un tamaño de población $R=1000$ individuos, una probabilidad de cruce de 0.8, una probabilidad de mutación de 0,01, un valor de la métrica $p=2$, un número de generaciones $G=500$ y un valor del número de Soluciones de Precisión $SP=10$. Los valores del tamaño N de la población de individuos no-dominada fueron siempre inferiores al tamaño (31) del frente real de Pareto y se siguió la secuencia $N=10,15,20,25,30$ individuos.

La métrica relativa $C(A,B)$, se empleó como medida de evaluación de resultados obtenidos con CMOEA&TOPSIS y TOPSISGA (A =frente alcanzado con TOPSISGA y B =frente alcanzado con CMOEA&TOPSIS). Las métricas usadas para la comparación de CMOEA&TOPSIS y NSGAI fueron la métrica relativa $C(A,B)$ (A =frente alcanzado con NSGAI y B =frente alcanzado con CMOEA&TOPSIS) y la métrica del hipervolumen H (se utilizó como punto de referencia el 40.00, 40.00).

Resultados visuales de CMOEA&TOPSIS

Las figuras 5.17, 5.18, 5.19, 5.20, 5.21 y 5.22 muestran que CMOEA&TOPSIS, encuentra un amplio conjunto de soluciones bien distribuido, pero acentúa el número de soluciones localizadas en aquella región que resulta de interés para el decisor (la RP). También se observa cómo la RP explorada por el algoritmo, varía al modificar los valores de los pesos o preferencias del decisor. Por último, en todas las figuras se observa el preorden (en base a TOPSIS) inducido (detallado por las seis primeras soluciones).

Comparativa CMOEA&TOPSIS y TOPSISGA: métrica C

Los resultados (media después de diez experimentos) de la comparación en base a la métrica relativa C, entre CMOEA&TOPSIS y TOPSISGA se muestran en la tabla 5.3. Se observa que para tamaños pequeños del fichero de soluciones no-dominadas ($N=10,15$ individuos) o tamaños próximos al número de soluciones del frente real de Pareto ($N=30$ individuos), los valores $C(A,B)$ y $C(B,A)$ son similares. Sin embargo, el porcentaje $C(A,B)$ de soluciones obtenidas con CMOEA&TOPSIS (frente B) dominadas por las soluciones logradas con TOPSISGA (frente A) es menor para tamaños intermedios. La figura 5.23 visualiza gráficamente y ayuda a una mejor interpretación de los resultados de la tabla 5.3.

La figura 5.24 izquierda, visualiza los frentes logrados por CMOEA&TOPSIS ($N=15$) y TOPSISGA ($N=10$) ($w_1=w_2=0.5$ para ambos algoritmos).

Comparativa CMOEA&TOPSIS y NSGAI: métricas C y S

La tabla 5.4 muestra los valores obtenidos al comparar los frentes alcanzados con CMOEA&TOPSIS y NSGAI con la métrica relativa $C(A,B)$. Los resultados dejan ver que el porcentaje (media después de diez experimentos) de soluciones obtenidas con CMOEA&TOPSIS (frente B) dominadas por las soluciones logradas con NSGAI (frente A) es menor para tamaños $N=10,15$ individuos de la población no-dominada. Esta diferencia desaparece cuando el tamaño N se hace mayor. La figura 5.25 visualiza gráficamente los resultados de la tabla 5.4.

La tabla 5.5 muestra los valores (media después de diez experimentos) obtenidos al comparar los conjuntos de soluciones alcanzados con CMOEA&TOPSIS y NSGAI en base al hipervolumen H (métrica S). CMOEA&TOPSIS parece tener una ligera ventaja con respecto a NSGAI para tamaños pequeños de la población no-dominada ($N=10,15$), pero desaparece para tamaños mayores de N . La figura 5.26 visualiza gráficamente los resultados de la tabla 5.5.

Una representación gráfica de los frentes obtenidos por CMOEA&TOPSIS ($N=15$, $w_1=w_2=0.5$) y NSGAI ($N=15$) se muestra en la figura 5.24 derecha.

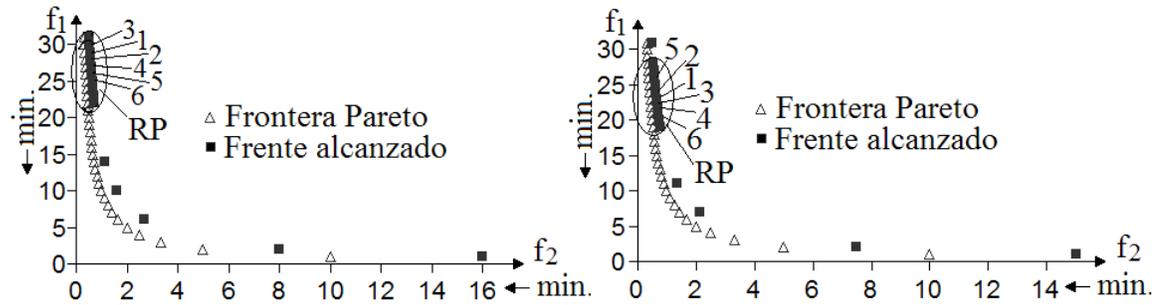


Figura 5.17: Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS: N=15, SP=10, $w_1=0.0001$ y $w_2=0.9999$ (izquierda) y $w_1=0.001$ y $w_2=0.999$ (derecha).

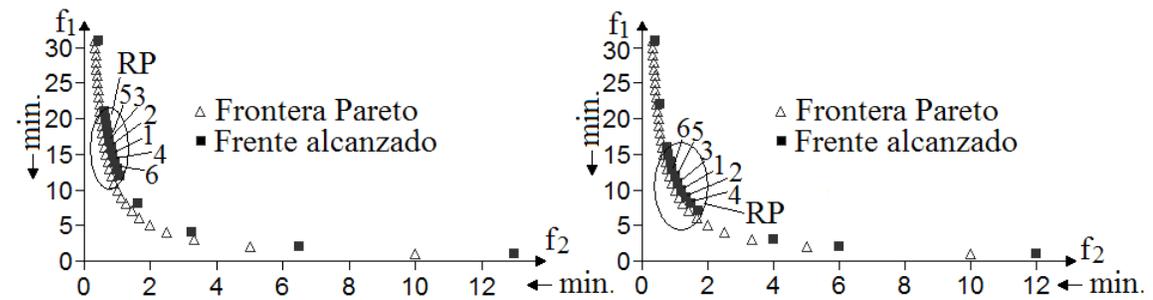


Figura 5.18: Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS: N=15, SP=10, $w_1=0.01$ y $w_2=0.99$ (izquierda) y $w_1=0.09$ y $w_2=0.91$ (derecha).

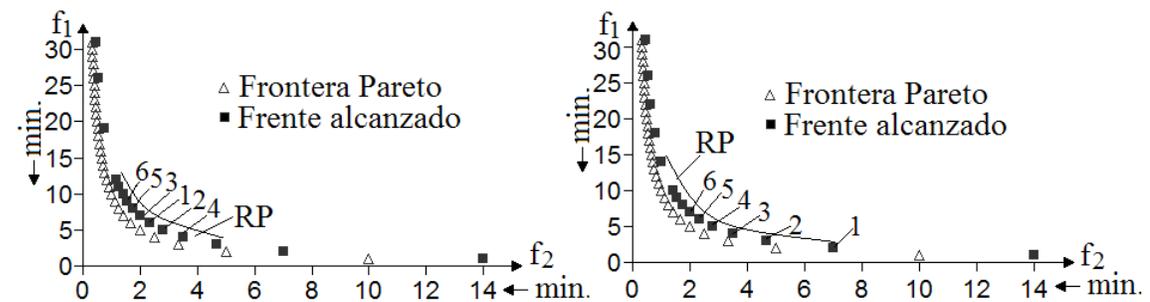


Figura 5.19: Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS: N=15, SP=10, $w_1=0.5$ y $w_2=0.5$ (izquierda) y $w_1=0.99$ y $w_2=0.01$ (derecha).

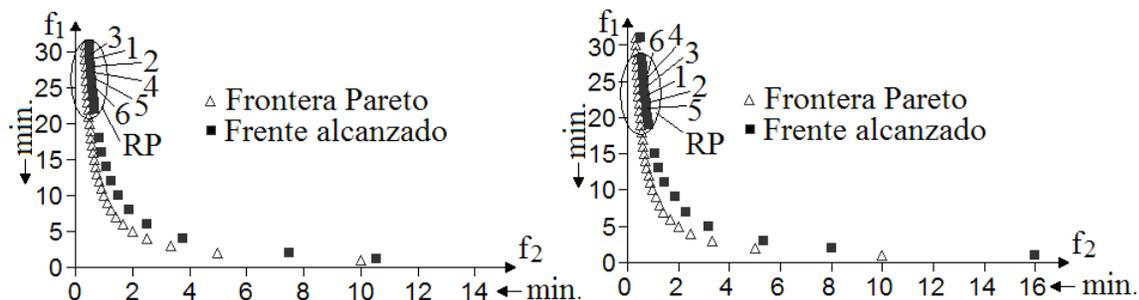


Figura 5.20: Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS: $N=20$, $SP=10$, $w_1=0.0001$ y $w_2=0.9999$ (izquierda) y $w_1=0.001$ y $w_2=0.999$ (derecha).

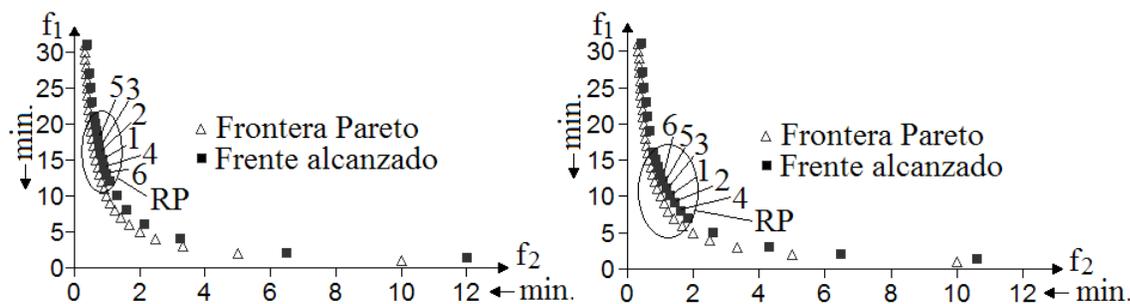


Figura 5.21: Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS: $N=20$, $SP=10$, $w_1=0.01$ y $w_2=0.99$ (izquierda) y $w_1=0.09$ y $w_2=0.91$ (derecha).

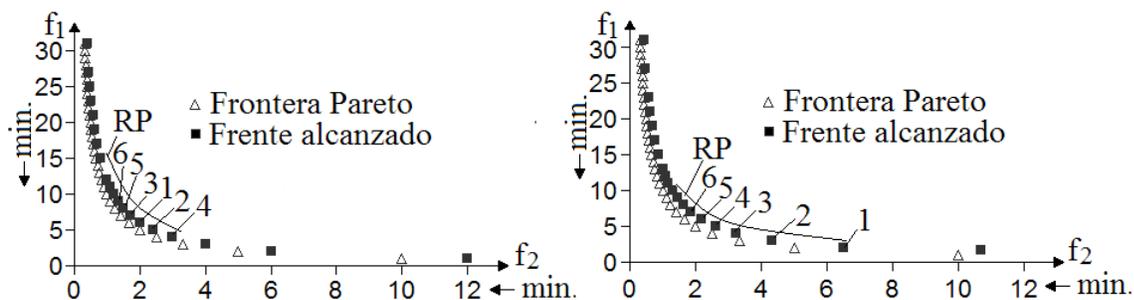


Figura 5.22: Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS: $N=20$, $SP=10$, $w_1=0.5$ y $w_2=0.5$ (izquierda) y $w_1=0.99$ y $w_2=0.01$ (derecha).

Tabla 5.3: Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes alcanzados con CMOEA&TOPSIS (frente B) y TOPSISGA (frente A).

	N=10	N=15	N=20	N=25	N=30
C(A,B)	32	32	32	30.4	39.34
C(B,A)	31.5	33.3	48	40	40

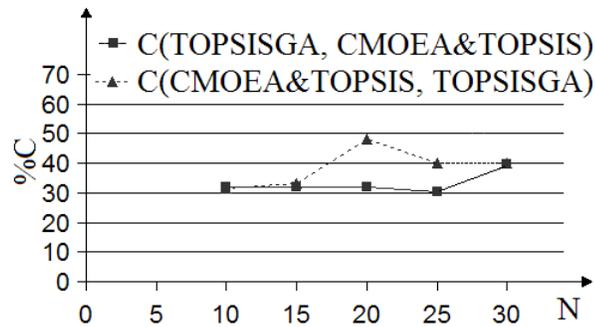


Figura 5.23: Visualización gráfica de la tabla 5.3.

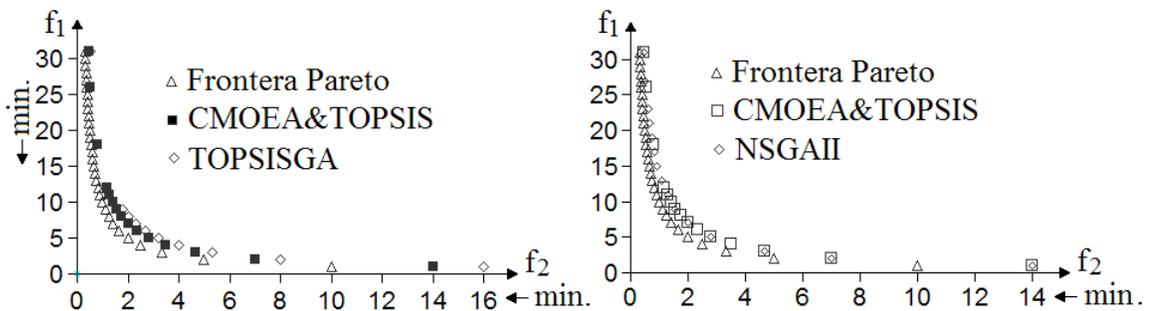


Figura 5.24: Comparación de los frentes alcanzados por CMOEA&TOPSIS (se omite la clasificación TOPSIS inducida) ($N=15$, $SP=10$ y $w_1=w_2=0.5$) con TOPSISGA ($N=10$ y $w_1=w_2=0.5$) (izquierda) y con NSGAI ($N=15$) (derecha).

Tabla 5.4: Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes alcanzados con CMOEA&TOPSIS (frente B) y NSGAI (frente A).

	N=10	N=15	N=20	N=25	N=30
C(A,B)	10	8	20	40	40.4
C(B,A)	22	26.6	20	40.2	40

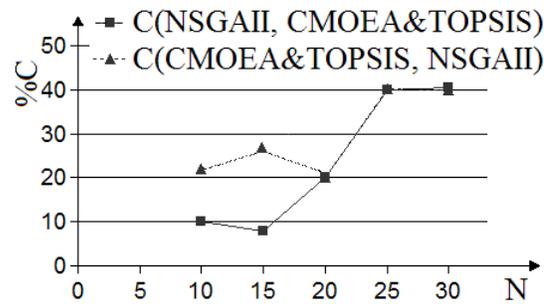


Figura 5.25: Visualización gráfica de la tabla 5.4.

Tabla 5.5: Resultados obtenidos en la comparativa de los frentes alcanzados con CMOEA&TOPSIS y NSGAI, en base al hipervolumen H (métrica S).

Método	N=10	N=15	N=20	N=25	N=30
CMOEA&TOPSIS	1491.6	1496.55	1494.3	1497.27	1495.98
NSGAI	1483.23	1484.03	1494.21	1496.01	1497.33

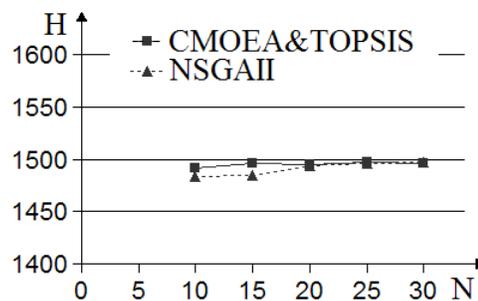


Figura 5.26: Visualización gráfica de la tabla 5.5.

5.4.4 Conclusiones

Se propone CMOEA&TOPSIS, un AEMO combo que ofrece en un mismo lote las mejores prestaciones de los mejores AEMOs a posteriori y priori. El algoritmo propuesto, aplica conjuntamente los operadores de crowding y similarity, en aquel frente de soluciones que no cabe completamente en el fichero de soluciones no-dominadas de tamaño N . El resultado final son dos conjuntos de soluciones: el primero, de tamaño el número SP de soluciones elegido por el decisor, se caracteriza por tener una alta y constante densidad de soluciones, el segundo esboza ampliamente y con buena distribución el resto del frente óptimo. Los experimentos llevados a cabo, muestran que cuando se comparan TOPSISGA y CMOEA&TOPSIS, este último consigue un mejor equilibrio exploración-explotación (hay más diversidad), pues en general se obtiene un menor porcentaje de soluciones alcanzadas con CMOEA&TOPSIS que son dominadas por las obtenidas con TOPSISGA (sólo para tamaños de N pequeños o próximos al tamaño real del frente, los valores de las métricas son parecidos). También, los resultados de comparar CMOEA&TOPSIS con NSGAI con la métrica C , son mejores para el algoritmo propuesto para valores del tamaño de la población no-dominada inferiores al tamaño de soluciones de la frontera real del problema, igualándose al incrementarse dicho tamaño. Resultados similares a la métrica C , son los obtenidos con la métrica S . Una cuestión importante a remarcar, es la similitud de los resultados al comparar los algoritmos con las métricas C y S , cuando el tamaño N de la población no-dominada tiende al tamaño real del frente de Pareto (31 soluciones para el problema ZDT5 que aquí utilizamos). Esto se explica por el hecho de que los operadores (similarity de TOPSISGA, crowding distance de NSGAI y similarity+crowding distance de CMOEA&TOPSIS) están grandemente limitados para valores de población no-dominada próximos o superiores a $N=31$. Por último, señalar que CMOEA&TOPSIS (de forma similar al método TOPSISGA) aborda la problemática P_γ de Roy [161][163](sección 1.6) e induce un preorden completo en las soluciones alcanzadas en la RP en base a la similaridad [45] a la solución ideal según el método TOPSIS.

5.5 CMOEA20-80&TOPSIS

5.5.1 Introducción

El algoritmo propuesto CMOEA&TOPSIS en la sección 5.4, define dos regiones en el frente de soluciones alcanzado. Una región (sobre la cual actúa el operador similarity), denominada Región de Precisión RP y caracterizada por su alta densidad de soluciones y preorden completo inducido en ellas (en base a TOPSIS [101]) y una segunda región (sobre la cual actúa el operador crowding distance), que esboza el

resto del frente de soluciones de manera amplia y bien distribuida. En cuanto al tamaño de la RP, éste es precisado por el decisor (parámetro: Soluciones de Precisión SP deseadas). Se observa que, si el valor de SP elegido por el decisor es igual al tamaño N de la población de soluciones no-dominadas, entonces el comportamiento de CMOEA&TOPSIS es idéntico al de TOPSISGA (sección 5.2). Por otro lado, si el valor SP adoptado es cero, entonces el comportamiento de CMOEA&TOPSIS es análogo al de NSGAI [51][55]. En los experimentos que se llevaron a cabo con CMOEA&TOPSIS (sección 5.4.3), se adoptó (sin ningún criterio en especial) un tamaño SP=10. La cuestión que se plantea ahora es: ¿Qué tamaño SP sería el idóneo elegir?

Una regla cuantitativa muy sencilla y que puede aplicarse para muchos aspectos de la vida (por ejemplo clasificaciones de inventarios, de clientes, etc.), es la conocida regla 20-80 de Pareto. Esta se fundamenta en los resultados obtenidos por Wilfredo Pareto [149] tras un estudio de la distribución de las rentas de los individuos que componen una nación. El resultado del mismo fue que un gran porcentaje de las rentas (el 80%), estaba concentrado en manos de un pequeño porcentaje de individuos (el 20%) (figura 5.27). Esta regla se conoce como la ley de Pareto y se generaliza según: *hay unos pocos valores críticos y muchos insignificantes, es en los valores críticos y no en los insignificantes, donde deben concentrarse los recursos.*

El algoritmo CMOEA20-80&TOPSIS propuesto en esta sección, obtiene (como lo hace CMOEA&TOPSIS) un amplio y bien distribuido frente óptimo de soluciones (acentuando el número de soluciones en la región de interés), pero utiliza la regla de Pareto para estimar en cada iteración, el valor del parámetro SP.

5.5.2 Algoritmo propuesto

El algoritmo CMOEA20-80&TOPSIS que se propone aquí, posee la misma estructura que CMOEA&TOPSIS (ver la sección 5.4 y la figura 5.1), pero difiere en como se calcula el tamaño SP de la RP. En CMOEA&TOPSIS, es el decisor quien decide su valor. En CMOEA20-80&TOPSIS el tamaño SP es evaluado automáticamente en cada iteración según la proposición que sigue:

Proposición 1: *El tamaño SP de la RP, debe ser estimado como un porcentaje (20%) del tamaño del frente de soluciones.*

Así, cuando las soluciones de un frente F_i no entran completamente en la nueva población P^{t+1} , se aplican para mantener el tamaño N (aunque las soluciones extremas tienen preferencia) los operadores similarity y crowding distance en los siguientes términos: el frente F_i es clasificado por similaridad a la solución ideal y las soluciones con mayor valor de similaridad pasan a formar parte de la nueva población hasta un valor máximo del 20% del tamaño del frente F_i , si la población continua

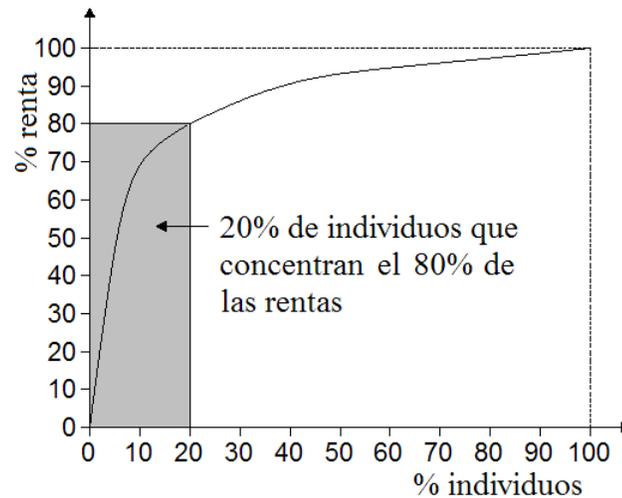


Figura 5.27: Distribución de las rentas de una nación y valores críticos según la regla 20-80 de Pareto.

sin completarse, se eligen del resto de soluciones del frente F_i aquellas que resultan ser las mejores en la clasificación por crowding distance. Luego, en la selección de soluciones por torneo para la generación de la nueva población de descendientes, las soluciones preferidas se escogen según: (1) solución con menor valor de rango, (2) soluciones extremas, (3a) solución con mayor valor de similaridad (si la selección se efectúa entre soluciones que han sido clasificadas por similaridad), (3b) al azar (si la selección se efectúa entre soluciones que han sido clasificadas por distancia crowding) y (3c) solución clasificada por similaridad (si la selección se efectúa entre una solución clasificada por similaridad y otra clasificada por distancia crowding).

5.5.3 Resultados experimentales

CMOEA20-80&TOPSIS se comparó aquí con NSGAI, en base a las métricas C y S. En todos los experimentos se usó un tamaño de población $R=1000$ individuos, una probabilidad de cruce de 0.8, una probabilidad de mutación de 0,01, un valor de la métrica $p=2$ y un número de generaciones $G=500$. Los valores del tamaño N de la población de individuos no-dominada fueron siempre inferiores al tamaño (31) del frente real de Pareto y se siguió la secuencia $N=10,15,20,25,30$ individuos. Para la métrica S, se utilizó el punto (40.00, 40.00) como punto de referencia.

Tabla 5.6: Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes alcanzados con CMOEA20-80&TOPSIS (frente B) y NSGAI (frente A).

	N=10	N=15	N=20	N=25	N=30
C(A,B)	20	17.34	20	40	40.1
C(B,A)	30	41.33	40.2	40.1	40.3

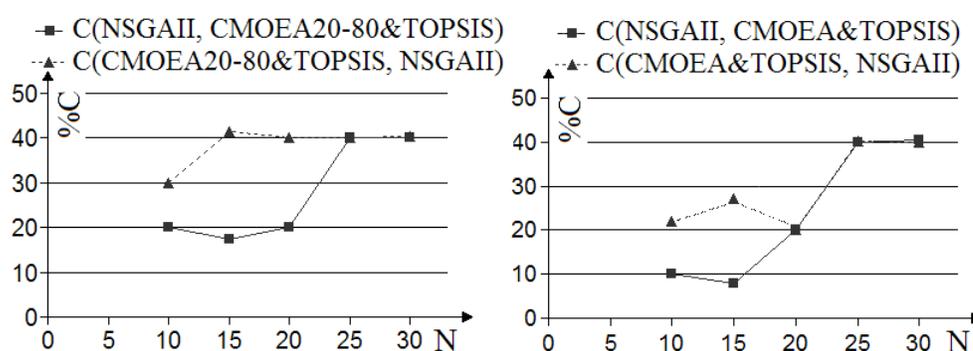


Figura 5.28: Visualización gráfica de las tablas 5.6 (izquierda) y 5.4 (derecha).

Tabla 5.7: Resultados obtenidos en la comparativa de los frentes alcanzados con CMOEA20-80&TOPSIS y NSGAI, según el hipervolumen H (métrica S).

Método	N=10	N=15	N=20	N=25	N=30
CMOEA2080&TOPSIS	1492.13	1499.5	1499.72	1496.83	1498.26
NSGAI	1483.23	1484.03	1494.21	1496.01	1497.33

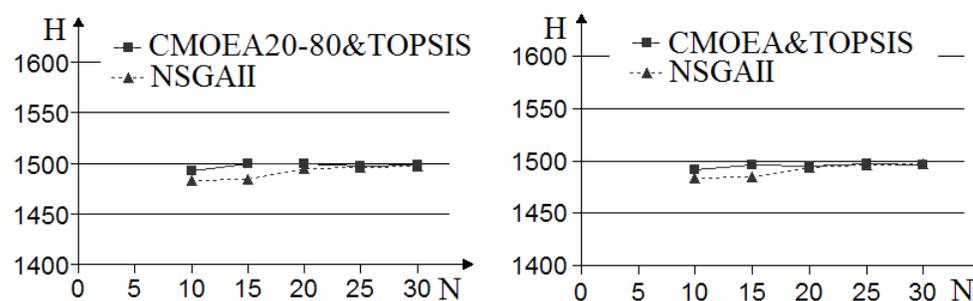


Figura 5.29: Visualización gráfica de las tablas 5.7 (izquierda) y 5.5 (derecha).

Comparativa CMOEA20-80&TOPSIS y NSGAI: métricas C y S

La tabla 5.6 muestra los valores (media después de diez experimentos) obtenidos al comparar los frentes alcanzados con CMOEA20-80&TOPSIS y NSGAI con la métrica relativa C. Se observa que para tamaños $N=10,15,20$ individuos de la población no-dominada, el valor de la métrica C es menor (luego mejor) para CMOEA20-80&TOPSIS que para NSGAI. Para tamaños N próximos al tamaño real del frente de Pareto, los resultados son similares. La figura 5.28 izquierda, visualiza gráficamente los resultados de la tabla 5.6, además, para mejor valorar las diferencias de CMOEA20-80&TOPSIS (supera a NSGAI para $N=10,15,20$ individuos) con CMOEA&TOPSIS (supera a NSGAI para $N=10,15$ individuos), la figura 5.28 derecha, reproduce los resultados gráficos (figura 5.25) obtenidos con CMOEA&TOPSIS en la sección 5.4.3.

La tabla 5.7 muestra los valores (media después de diez experimentos) obtenidos al comparar los conjuntos de soluciones alcanzados con CMOEA20-80&TOPSIS y NSGAI con la métrica S (hipervolumen H). Como con la métrica C, CMOEA20-80&TOPSIS tiene una pequeña ventaja con NSGAI para tamaños $N=10,15,20$ de la población no-dominada. La figura 5.29 izquierda, visualiza gráficamente los resultados de la tabla 5.7 y la figura 5.29 derecha, reproduce la figura 5.26 obtenida en la sección 5.4.3 con CMOEA&TOPSIS a partir de la tabla 5.5.

5.5.4 Conclusiones

Se propone CMOEA20-80&TOPSIS, un AEMO que tiene la misma estructura que CMOEA&TOPSIS (sección 5.4), pero el valor SP o Soluciones de Precisión que se quieren en la región de interés, es estimado de manera automática en cada iteración del algoritmo aplicando la regla 20-80 de Pareto [149]. El algoritmo propuesto, obtiene un amplio y bien distribuido frente óptimo de soluciones, pero resalta con una elevada densidad de soluciones (e induce en ellas un preorden completo en base a la similaridad [45] a la solución ideal) la región de interés. Además, los resultados en base a las métricas C y S, son mejores con CMOEA20-80&TOPSIS que los obtenidos con NSGAI y CMOEA&TOPSIS para tamaños $N=10,15,20$ individuos de la población no-dominada. Al incrementarse el tamaño N , los resultados para ambos algoritmos se igualan, pues de nuevo remarcamos que para tamaños N de la población no-dominada próximos o superiores al tamaño real (31) del frente de Pareto del problema, los operadores crowding distance de NSGAI y similarity+crowding distance de CMOEA20-80&TOPSIS están muy limitados.

Capítulo 6

Aplicaciones a dos Problemas Multiobjetivo Discretos

6.1 Introducción

En este capítulo se presentan los resultados de aplicar los AEMOs implementados en el capítulo 5 de contribuciones de esta tesis, a dos problemas de optimización multiobjetivo discretos.

El primero de los problemas, es el problema de la Mochila Multiobjetivo en variables binarias (0-1MOKP¹). Se trata de un problema muy conocido y estudiado en la comunidad multiobjetivo, que modela numerosas aplicaciones reales en campos como la economía, las finanzas y en cualquier problema que se ajuste al modelo descrito por la ecuación (6.2.1). F. Mallor-Giménez et al. [128], utilizan el MOKP para tratar un problema del mundo real (sector energético), en el que se necesita un análisis de la viabilidad y dimensionamiento de un sistema de almacenamiento y transformación de energía. El 0-1MOKP es fácil de implementar pero en la práctica resulta difícil de resolver dada su naturaleza NP-difícil. Numerosos AEMOs han sido utilizados para resolver el 0-1MOKP [105][107][135][174][219].

El segundo de los problemas, consiste en un problema multiobjetivo de Optimización en Diseño de Sistemas de Seguridad (ODSS). El diseño de sistemas, es una disciplina de la ingeniería, que debido al incremento de la complejidad de los sistemas y a la necesidad de tener que obtener diseños óptimos más que simplemente adecuados, ha experimentado en la última década una importante evolución. En estos sistemas se debe hacer un especial énfasis al diseño de Sistemas de Seguridad (un fallo en el sistema puede implicar daño a personas), considerando alternativas de diseño y/o estrategias de mantenimiento, con la finalidad de desarrollar una optimización global. En ambos casos, métodos multiobjetivo son especialmente adecuados, pues los

¹Siglas en inglés de Multiobjective Knapsack Problem.

decisores están especialmente interesados en tomar sus decisiones en base a diferentes objetivos en conflicto. Como aplicación práctica de ODSS, se ha usado el Sistema de Contención por Inyección en Spray (SCIS) de una central nuclear de generación eléctrica. Este problema es también de naturaleza NP-difícil y ha sido ampliamente estudiado en [78][89][91]. Se trata de un problema clásico en las líneas de investigación seguidas en la división de Computación Evolutiva y Aplicaciones (CEANI) del Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (IUSIANI) de la Universidad de Las Palmas de Gran Canaria (ULPGC).

6.2 Problema de la Mochila Multiobjetivo

6.2.1 Descripción del problema

En general, en el problema de la Mochila Multiobjetivo, se consideran n mochilas con sus correspondientes capacidades c_i , así como un conjunto de m objetos con diferentes pesos w_{ij} y beneficios b_{ij} . El problema consiste en encontrar un subconjunto de objetos de forma que el beneficio sea máximo pero sin sobrepasar las capacidades límite (pesos máximos) de cada mochila. Formalmente, el problema puede ser definido de la siguiente forma:

$$\left\{ \begin{array}{ll} \max. & f_i(x) = \sum_{j=1}^m b_{ij}x_j \quad i = 1, 2, \dots, n \\ \text{tal que} & \sum_{j=1}^m w_{ij}x_j \leq c_i \quad x_j \in \{0, 1\} \end{array} \right. \quad (6.2.1)$$

donde:

n = número de mochilas

m = número de objetos

x_j = variable de decisión (objetos que se seleccionan)

w_{ij} = peso del objeto j en la mochila i

b_{ij} = beneficio del objeto j en la mochila i

c_i = capacidad de la mochila i

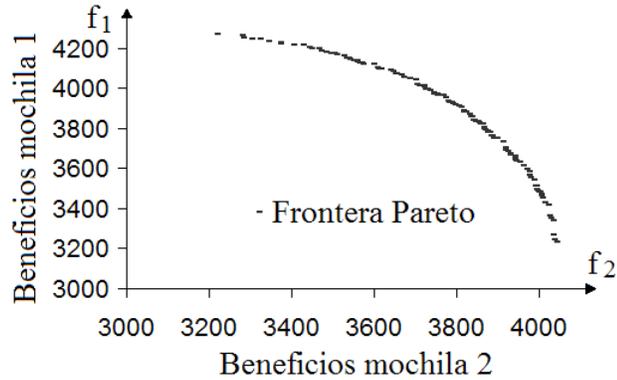


Figura 6.1: Frontera de Pareto del problema de la Mochila Multiobjetivo (2 mochilas).

Tabla 6.1: Pesos y beneficios adoptados para los objetos en la mochila 1 y en la mochila 2.

Pesos mochila 1																								
94	74	77	74	29	11	73	80	81	82	75	42	44	57	20	20	99	95	52	81	68	16	79	30	16
90	21	49	70	78	77	21	84	19	65	38	25	43	99	75	80	10	44	26	21	74	20	22	81	89
15	35	24	16	43	75	25	76	48	75	15	23	10	81	81	67	58	77	49	16	65	74	14	41	74
74	17	12	95	29	75	61	59	37	75	90	17	79	15	88	76	93	98	80	33	39	96	71	39	49
Pesos mochila 2																								
55	10	97	73	69	23	62	47	90	62	96	88	95	61	94	16	91	61	27	18	96	77	14	36	17
56	83	41	52	69	97	45	94	45	47	28	82	13	82	39	28	41	55	12	50	32	97	87	36	11
20	37	87	91	19	22	89	54	20	78	52	35	18	96	54	10	39	17	51	40	25	84	54	31	97
37	63	39	60	87	63	37	13	31	84	90	84	57	21	64	63	21	95	83	81	45	68	89	100	14
Beneficios mochila 1																								
57	94	59	83	82	91	42	84	85	18	94	18	31	27	31	42	58	57	55	97	79	10	34	100	98
45	19	77	56	25	60	22	84	89	12	46	20	85	42	94	20	65	27	34	27	91	17	56	23	89
18	11	91	79	14	99	45	73	81	96	51	96	63	40	93	87	71	54	74	15	32	57	70	62	12
71	57	97	48	33	42	15	39	91	17	63	81	49	60	90	87	25	15	30	76	76	53	59	40	59
beneficios mochila 2																								
20	19	20	66	48	100	13	87	62	73	53	79	17	93	78	22	85	86	56	56	44	86	94	93	57
31	20	35	70	79	58	24	84	12	17	43	35	47	92	38	93	50	27	100	36	30	23	22	56	73
55	32	75	42	82	80	55	48	93	28	26	42	96	93	16	39	46	80	24	87	37	73	81	38	98
13	91	85	17	59	58	56	93	66	64	17	10	33	28	97	25	42	17	23	37	46	52	33	26	90

6.2.2 Datos adoptados

Los datos que se han adoptado han sido los de Zitzler [219] para dos mochilas: capacidad mochila 1 = 2732, capacidad mochila 2 = 2753, número de objetos = 100 con pesos y beneficios los que se muestran en la tabla 6.1. El verdadero frente de Pareto es conocido (121 soluciones) y se muestra en la figura 6.1.

6.2.3 Configuración de parámetros

En todos los experimentos se utilizó un tamaño de la población R de 200 individuos, una probabilidad de cruce de 0.8, una probabilidad de mutación de 0.01, un valor $p=2$ para la métrica p en la ecuación (1.10.1) y el número máximo de generaciones (G) fue de 500. El tamaño N del fichero de soluciones no-dominadas, fue cambiado progresivamente según las secuencias $N=10,15,20,30,40,50,100$.

6.2.4 Métricas

Las métricas utilizadas para la evaluación y comparación de resultados fueron la métrica relativa $C(A,B)$ [222] para medir el porcentaje (media después de diez experimentos) de soluciones de B que son dominadas por al menos una solución de A (menor valor de $C(A,B)$, mejor para B y peor para A) (ver la sección 4.5.1.) y la métrica S [222] (mayor valor de S , mejor) para medir el tamaño (media después de diez ensayos) de la región dominada por el frente de Pareto y limitada por un punto de referencia (se adoptó el punto de coordenadas 0.00, 0.00) (ver la sección 4.5.2).

6.2.5 Análisis de resultados con TOPSISGA

Para evaluar la eficacia de TOPSISGA, éste se comparó con NSGAII [51][55] en base a la métrica C . Los resultados obtenidos se detallan en las subsecciones posteriores.

Las figuras 6.2 y 6.3 muestran los resultados obtenidos con TOPSISGA ($N=10$) para diferentes valores de los pesos o preferencias w_1 y w_2 . Las numeraciones corresponden a la clasificación TOPSIS [101] del frente alcanzado (preorden completo acorde a las preferencias del decisor). Se observa a partir de ambas figuras, cómo la región de exploración y el preorden final de las soluciones, cambian cuando el decisor varía los valores de los pesos.

En la figura 6.4 se visualizan los frentes logrados con TOPSISGA y NSGAII (para $N=10$). Remárquese que TOPSISGA se concentra en una particular región de la frontera eficiente, mientras que NSGAII encuentra un amplio y bien distribuido conjunto eficiente de soluciones. De otro lado, es evidente que es imposible obtener una similar clasificación a la obtenida por TOPSISGA a partir del conjunto final obtenido con NSGAII y viceversa.

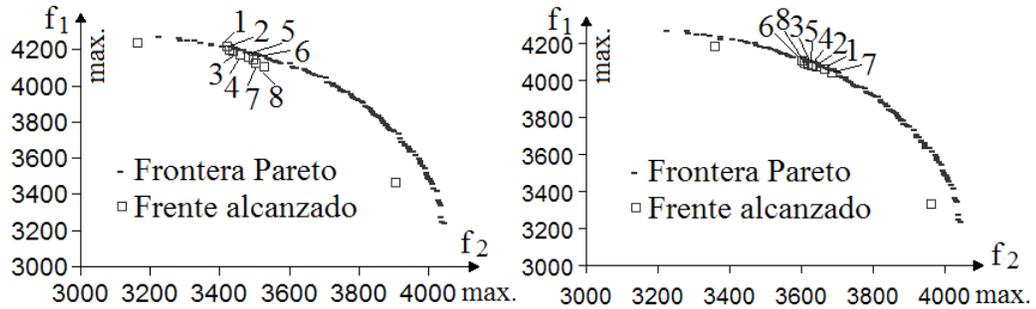


Figura 6.2: Frente óptimo y preorden logrado por TOPSISGA: $N=10$, $w_1=0.9$, $w_2=0.1$ (izquierda) y $w_1=0.75$, $w_2=0.25$ (derecha) con el problema de la Mochila Multiobjetivo.

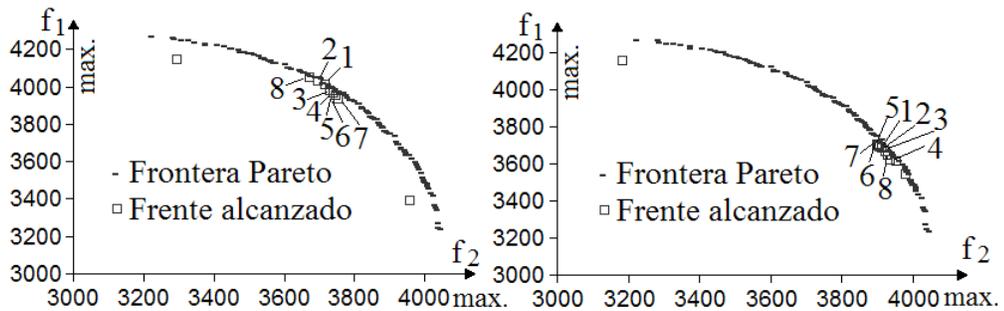


Figura 6.3: Frente óptimo y preorden logrado por TOPSISGA: $N=10$, $w_1=w_2=0.5$ (izquierda) y $w_1=0.1$, $w_2=0.9$ (derecha) con el problema de la Mochila Multiobjetivo.

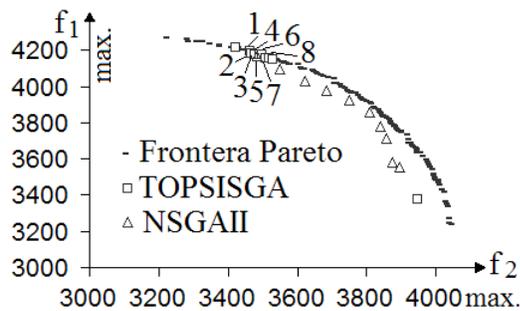


Figura 6.4: Visualización de los frentes alcanzados ($N=10$ individuos) por TOPSISGA ($w_1=0.9$ y $w_2=0.1$) y NSGAI.

Tabla 6.2: Valores en tanto por ciento de la métrica relativa $C(A,B)$, obtenidos al comparar los frentes alcanzados con TOPSISGA (frente B) y NSGAI (frente A).

	N=10	N=15	N=20	N=30	N=40	N=50	N=100
$C(A,B)$	4	10.6	14	28.9	37.38	41.5	41.8
$C(B,A)$	18	17.31	26	30.66	37.38	41.5	41

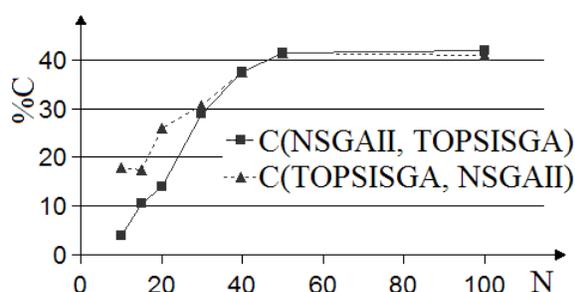


Figura 6.5: Vista gráfica de los valores de la tabla 6.2.

Comparativa TOPSISGA y NSGAI: métrica C

En la tabla 6.2, se muestran los resultados obtenidos al comparar los frentes alcanzados con TOPSISGA y NSGAI con la métrica relativa $C(A,B)$ (con B=frente alcanzado con TOPSISGA y A=frente alcanzado con NSGAI). Se observa, que el valor $C(A,B)$ o porcentaje de soluciones obtenidas con TOPSISGA dominadas por las soluciones alcanzadas con NSGAI es menor para valores $N=10,15,20$ individuos de la población no-dominada, desapareciendo esta diferencia al incrementar el valor N . La figura 6.5 visualiza los valores de la tabla 6.2.

6.2.6 Análisis de resultados con WP-TOPSISGA

Las figuras 6.6, 6.7, 6.8, 6.9 y 6.10, muestran los resultados visuales obtenidos al comparar WP-TOPSISGA y R-NSGAI [56], cuando $N=15$ individuos y valores del PF ($f_1=4097, f_2=3395$), ($f_1=4180, f_2=3395$), ($f_1=3806, f_2=3758$), ($f_1=3392, f_2=3836$) y ($f_1=3581, f_2=3930$) respectivamente. Nótese que para posiciones escoradas del PF hacia los extremos del frente, WP-TOPSISGA concentra mejor las soluciones en la región de interés del decisor región de interés (la R_d -PF). Para valores más centrados del PF (figura 6.8 con $N=15$ individuos como tamaño del conjunto de soluciones no-dominadas y como coordenadas del PF los valores $f_1=3806, f_2=3758$) o bien valores del PF escorados hacia los extremos pero cercanos a la frontera real de Pareto (figura

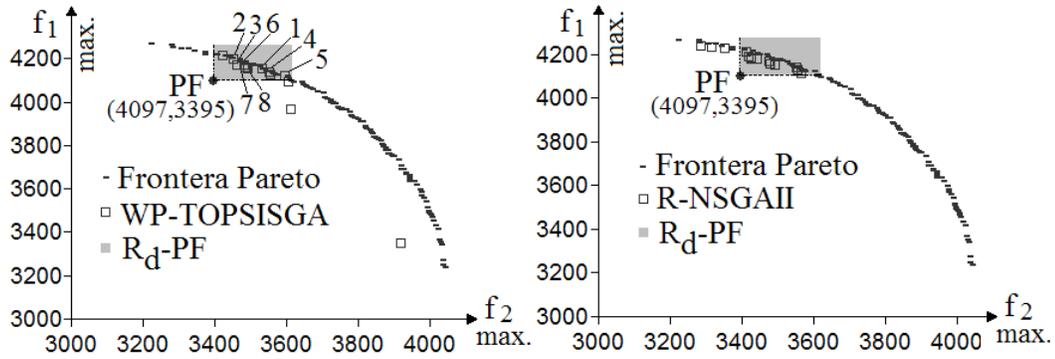


Figura 6.6: Frentes alcanzados (N=15) con WP-TOPSISGA y R-NSGAI.

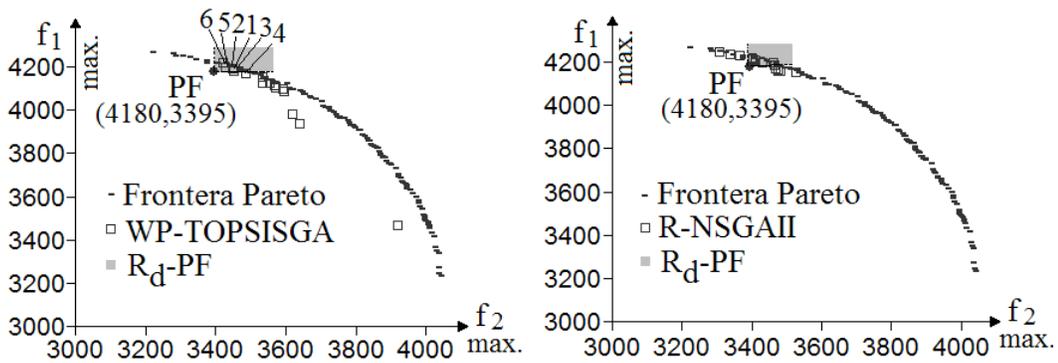


Figura 6.7: Frentes alcanzados (N=15) con WP-TOPSISGA y R-NSGAI.

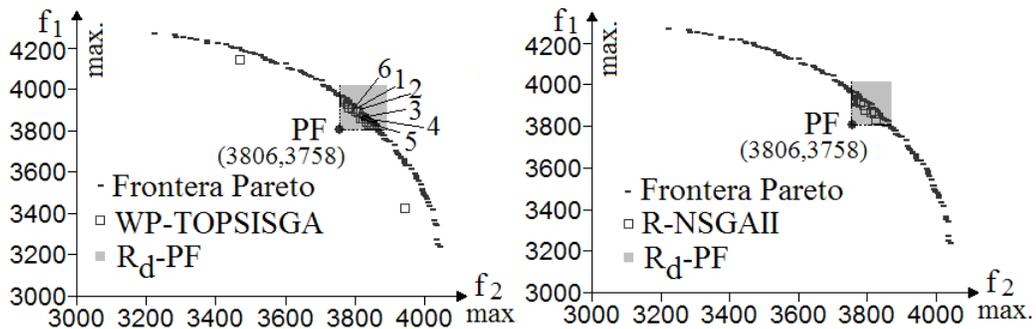


Figura 6.8: Frentes alcanzados (N=15) con WP-TOPSISGA y R-NSGAI.

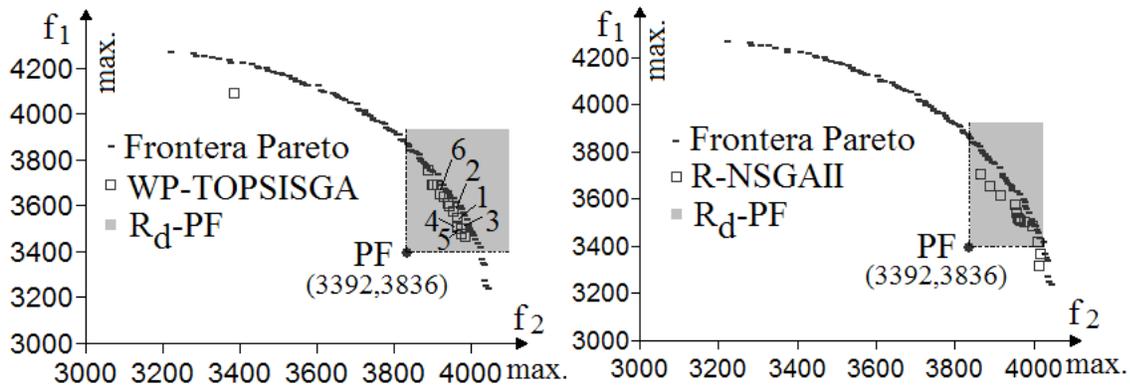


Figura 6.9: Frentes óptimos alcanzados ($N=15$ individuos) con WP-TOPSISGA y R-NSGAI.

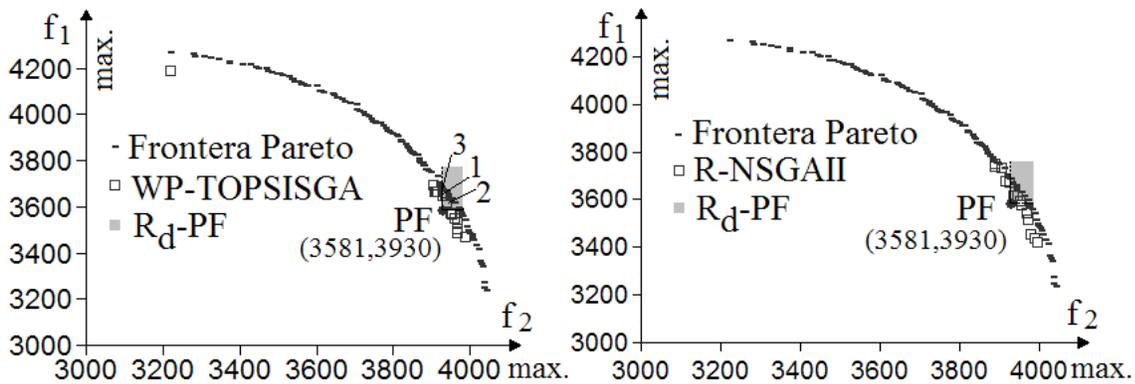


Figura 6.10: Frentes óptimos alcanzados ($N=15$ individuos) con WP-TOPSISGA y R-NSGAI.

Tabla 6.3: Valores en tanto por ciento de la métrica relativa C, obtenidos cuando se comparan los frentes alcanzados (en la R_d -PF) con WP-TOPSISGA (frente B) y R-NSGAI (frente A).

	N=15	N=20	N=30	N=40	N=50	N=100
($f_1=4097, f_2=3395$)						
C(A,B)	0	25.5	35.8	35.06	38.1	38.5
C(B,A)	53	36.9	35.1	36.14	39.01	38.3
($f_1=4180, f_2=3395$)						
C(A,B)	10	24.1	35.4	35.13	39.1	38.82
C(B,A)	48	36.2	34.9	35.91	39.32	38.22
($f_1=3806, f_2=3758$)						
C(A,B)	11.32	27.1	32.2	36.86	37.1	37.98
C(B,A)	12	26.6	31.8	36.05	37.01	38.1
($f_1=3392, f_2=3836$)						
C(A,B)	10	19.98	32.6	37.24	38.1	37.9
C(B,A)	44	29.96	33.3	36.16	39.01	38.71
($f_1=3581, f_2=3930$)						
C(A,B)	12.8	31.66	31.7	35.92	38.03	37.99
C(B,A)	13.3	33.32	31.3	36.3	37.25	37.36

6.10 con $N=15$ individuos como tamaño del conjunto de soluciones no-dominadas y como coordenadas del PF los valores $f_1=3581$, $f_2=3930$), los resultados visuales son parecidos con ambos algoritmos. Las numeraciones corresponden al preorden completo inducido (con WP-TOPSISGA) según TOPSIS.

Comparativa WP-TOPSISGA y R-NSGAI: métrica C

Los resultados de comparar WP-TOPSISGA y R-NSGAI en base a la métrica relativa C, se muestran en la tabla 6.3. Se observa que el valor C(A,B) o porcentaje de soluciones obtenidas con WP-TOPSISGA (frente B) dominadas por las soluciones alcanzadas con R-NSGAI (frente A) es menor (cuanto menos mejor) para valores de los PF=($f_1=4097, f_2=3395$), ($f_1=4180, f_2=3395$) y PF=($f_1=3392, f_2=3836$) (escorados hacia los extremos del frente), esta ventaja desaparece para valores más centrados (respecto el frente de Pareto) del PF (puntos $f_1=3806, f_2=3758$) o valores del PF (puntos $f_1=3581, f_2=3930$) escorado hacia el extremo inferior y cercano a la frontera real de Pareto. Las figuras 6.11, 6.12, 6.13, 6.14 y 6.15, visualizan gráficamente los valores de la tabla 6.3.

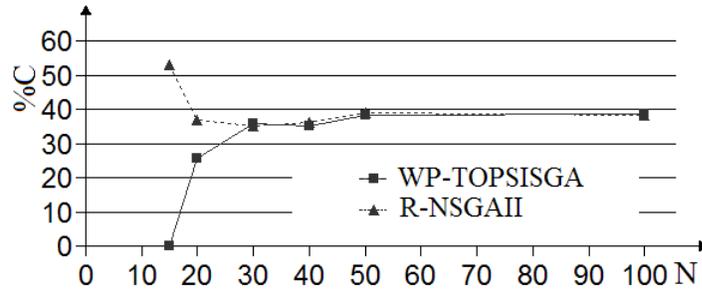


Figura 6.11: Resultados gráficos la tabla 6.3 con $PF=(f_1=4097, f_2=3395)$.

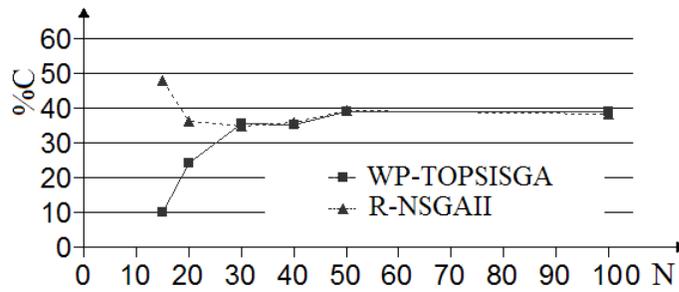


Figura 6.12: Resultados gráficos la tabla 6.3 con $PF=(f_1=4180, f_2=3395)$.

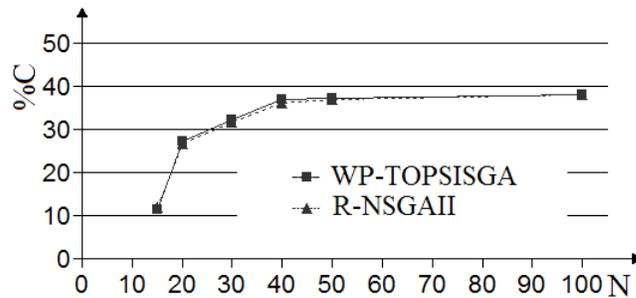


Figura 6.13: Resultados gráficos la tabla 6.3 con $PF=(f_1=3806, f_2=3758)$.

6.2.7 Análisis de resultados con CMOEA&TOPSIS

Para evaluar la eficacia de CMOEA&TOPSIS, éste se comparó con TOPSISGA (ver sección 5.2) y NSGAII [51][55] en base a las métricas C y S. Los resultados se detallan en las subsecciones posteriores.

Las figuras 6.16, 6.17, 6.18, 6.19, 6.20 y 6.21, ilustran los frentes óptimos de soluciones encontrados por CMOEA&TOPSIS y el preorden según TOPSIS [101] inducido en dichas soluciones. Se observa, cómo además de esbozar un frente amplio, el algoritmo acentúa el número de soluciones en la RP (para la cual se han precisado 10 SP). Remarcar, cómo la RP explorada difiere en función de los pesos manifestados por el decisor acorde a sus preferencias.

Comparativa CMOEA&TOPSIS y TOPSISGA: métrica C

Los resultados (ver tabla 6.4) reflejan, que la proporción de soluciones logradas con CMOEA&TOPSIS, dominadas por las alcanzadas con TOPSISGA es menor para tamaños reducidos del fichero de soluciones no-dominadas (N=15 individuos), pero la diferencia desaparece para otros tamaños de N (puntualizamos que para un tamaño N=10 e igual por tanto al número de Soluciones de Precisión SP que se han elegido para la RP, ambos algoritmos -CMOEA&TOPSIS y TOPSISGA- son equivalentes). La interpretación gráfica de los datos de la tabla 6.4 se muestran en la figura 6.22

Comparativa CMOEA&TOPSIS y NSGAII: métricas C y S

Las tablas 6.5, 6.6 y figuras 6.23, 6.24, recogen los resultados porcentuales de la métrica $C(A,B)$ (A=frente con NSGAII, B=frente con CMOEA&TOPSIS) cuando se comparan los frentes completos (todo el espacio) y frentes pertenecientes a la RP respectivamente. En la comparación de los frentes completos (tabla 6.5), se observa el menor porcentaje $C(A,B)$ (CMOEA&TOPSIS es mejor) para N=10 individuos, sin embargo, el porcentaje $C(B,A)$ (NSGAII es mejor) para tamaños de N comprendidos entre 15 y 50 individuos, por último para valores N superior o igual a 50 individuos los porcentajes $C(A,B)$ y $C(B,A)$ son parecidos. En la comparación de los conjuntos de soluciones obtenidos en la RP, los resultados de la métrica C son mucho mejores con CMOEA&TOPSIS que con NSGAII, para valores de N entre 10 y 50 individuos, a partir de N=50 los resultados son similares para ambos algoritmos.

La tabla 6.7 y figura 6.25 recogen los resultados (divididos por 10000 para facilitar la lectura) con la métrica S (hipervolumen H). Se observa (para tamaños N del fichero de soluciones no-dominadas entre 15 y 50 individuos), un ligero mejor valor de H para NSGAII que para CMOEA&TOPSIS.

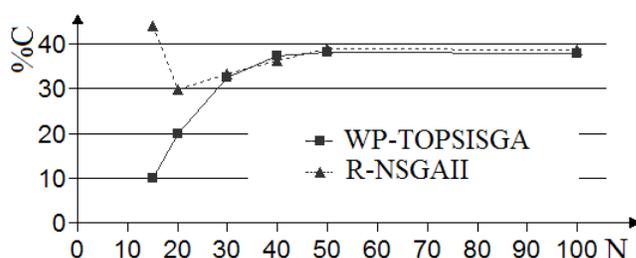


Figura 6.14: Resultados gráficos la tabla 6.3 con $PF=(f_1=3392, f_2=3836)$.

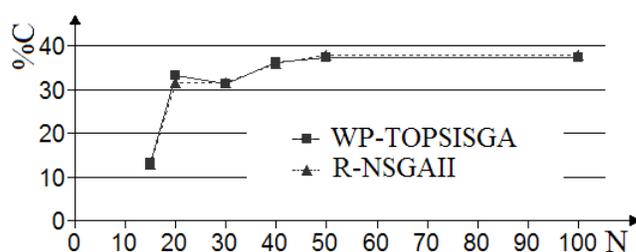


Figura 6.15: Resultados gráficos la tabla 6.3 con $PF=(f_1=3581, f_2=3930)$.

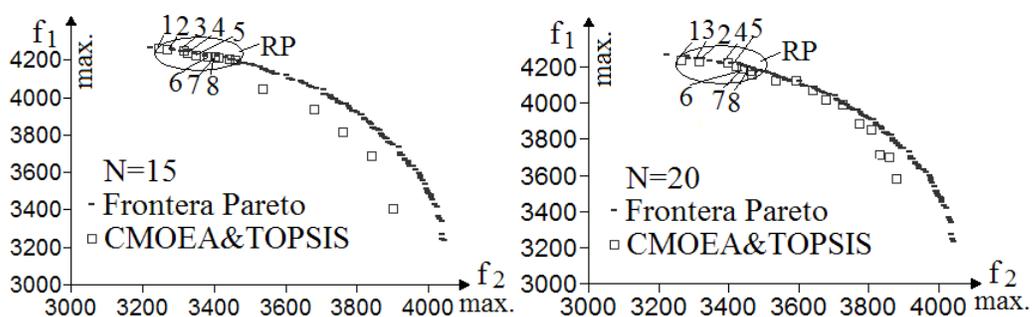


Figura 6.16: Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS para $SP=10$, $w_1=0.9999$ $w_2=0.0001$ y $N=15$ (izquierda), $N=20$ (derecha).

Tabla 6.4: Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes alcanzados con CMOEA&TOPSIS (frente B) y TOPSISGA (frente A).

	N=10	N=15	N=20	N=30	N=40	N=50	N=100
C(A,B)	28	30	31	30	39.9	41.8	42.3
C(B,A)	30	42.6	34	32.3	40.7	41.3	41.3

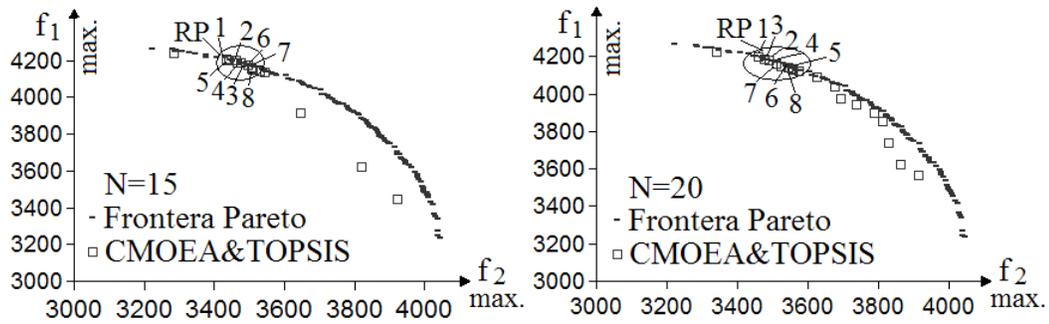


Figura 6.17: Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS para $SP=10$, $w_1=0.9$ $w_2=0.1$ y $N=15$ (izquierda), $N=20$ (derecha).

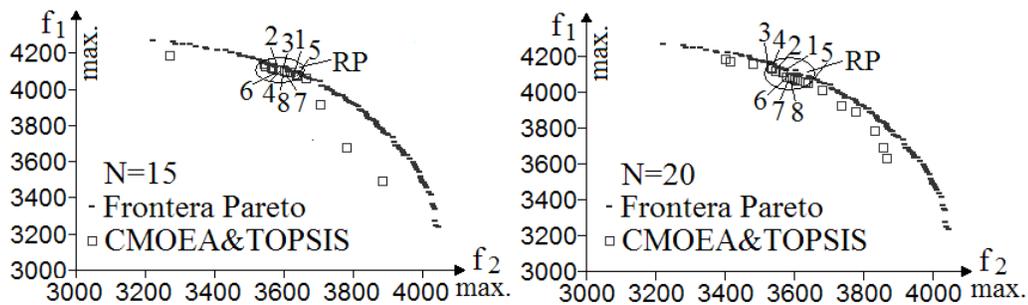


Figura 6.18: Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS para $SP=10$, $w_1=0.75$ $w_2=0.25$ y $N=15$ (izquierda), $N=20$ (derecha).

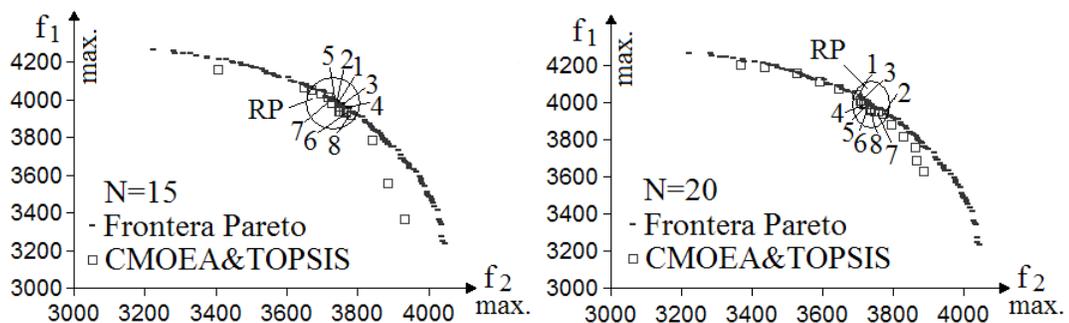


Figura 6.19: Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS para $SP=10$, $w_1=w_2=0.5$ y $N=15$ (izquierda), $N=20$ (derecha).

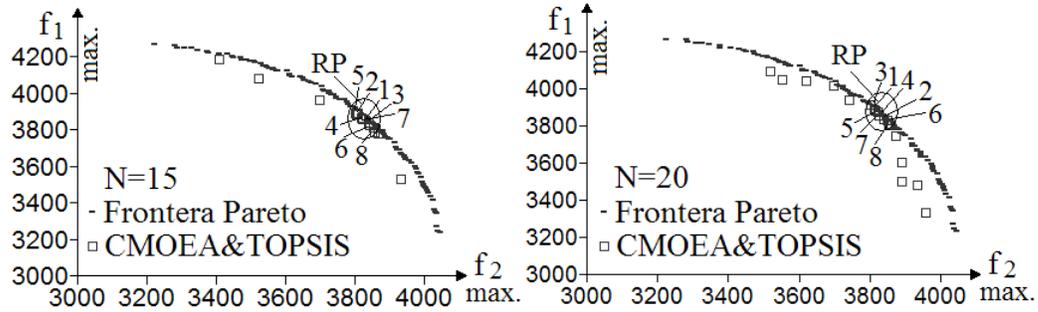


Figura 6.20: Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS para $SP=10$, $w_1=0.25$ $w_2=0.75$ y $N=15$ (izquierda), $N=20$ (derecha).

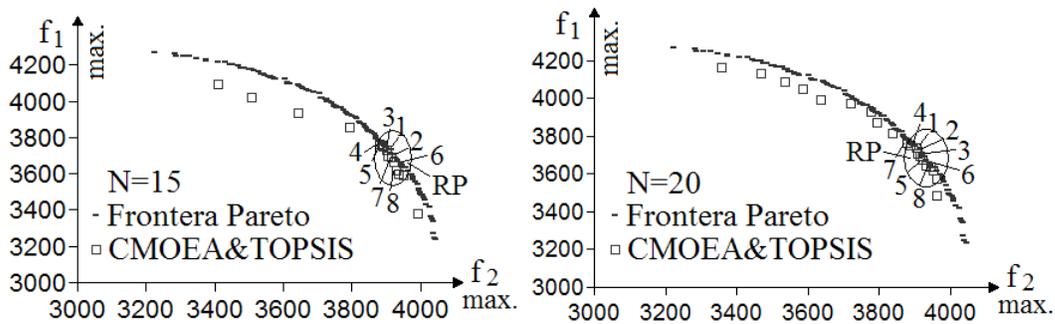


Figura 6.21: Frente óptimo y clasificación TOPSIS logrados con CMOEA&TOPSIS para $SP=10$, $w_1=0.1$ $w_2=0.9$ y $N=15$ (izquierda), $N=20$ (derecha).

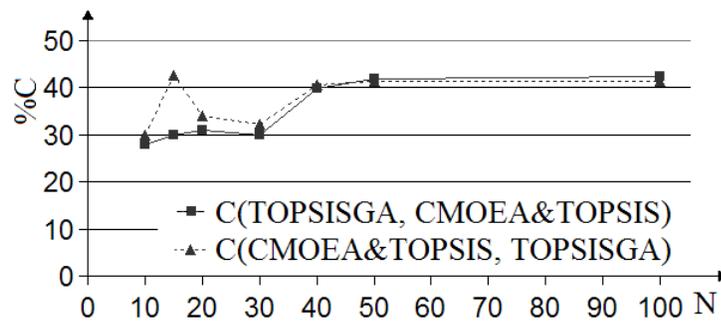


Figura 6.22: Visualización gráfica de la tabla 6.4.

Tabla 6.5: Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes completos alcanzados con CMOEA&TOPSIS (frente B) y NSGAI (frente A).

	N=10	N=15	N=20	N=30	N=40	N=50	N=100
C(A,B)	12	26.5	38	32.45	35.5	36.3	36.8
C(B,A)	22	16	27	30.5	35.24	35.9	36.9

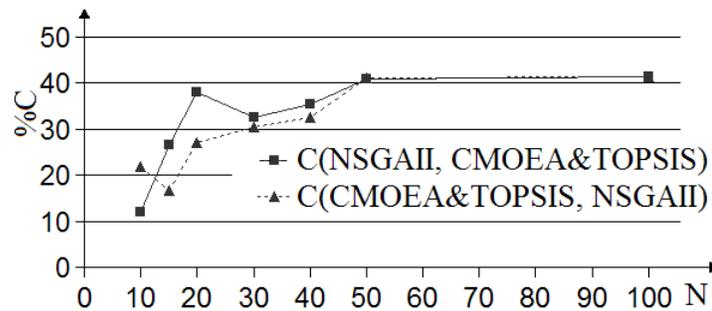


Figura 6.23: Visualización gráfica de la tabla 6.5.

Tabla 6.6: Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes (RP) alcanzados con CMOEA&TOPSIS (frente B) y NSGAI (frente A).

	N=10	N=15	N=20	N=30	N=40	N=50	N=100
C(A,B)	4	4	4	6.6	27.5	42.1	42.6
C(B,A)	80	66.6	71.6	48.3	61.32	41.9	42.9

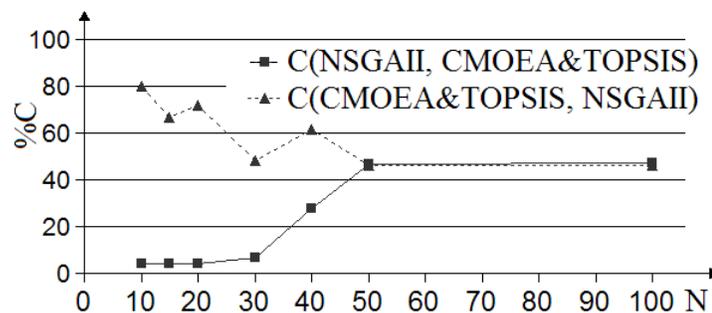


Figura 6.24: Visualización gráfica de la tabla 6.6.

6.2.8 Análisis de resultados con CMOEA20-80&TOPSIS

Comparativa CMOEA20-80&TOPSIS y NSGAI: métricas C y S

La tabla 6.8 muestra los valores obtenidos al comparar los frentes completos alcanzados con CMOEA20-80&TOPSIS y NSGAI con la métrica relativa C. La figura 6.26 izquierda (resultados visuales de la tabla 6.8) muestra comparándola con la figura 6.23 derecha (resultados visuales de la tabla 6.5 de comparar CMOEA&TOPSIS y NSGAI), que los valores de la métrica C son mejores para CMOEA20-80&TOPSIS con $N=10$ individuos pero mejores para NSGAI cuando el tamaño de N está comprendido entre 15 y 40 individuos, aunque las diferencias se han reducido.

En la comparación de los frentes logrados con CMOEA20-80&TOPSIS y NSGAI correspondientes a la RP, los resultados con la métrica C, son mejores para CMOEA20-80&TOPSIS que los obtenidos en la comparación de CMOEA&TOPSIS con NSGAI (ver la tabla 6.9 y figuras 6.27 izquierda y 6.27 derecha obtenida de la tabla 6.6).

Por último, cuando se comparan (a partir de tamaños $N=15$ individuos) los frentes obtenidos con CMOEA20-80&TOPSIS y NSGAI en base a la métrica S o hipervolumen H (ver la tabla 6.10 cuyos resultados están divididos por 10000 para facilitar la lectura y figuras 6.28 izquierda y 6.28 derecha obtenida de la tabla 6.7), se observa (para tamaños N entre 15 y 50 individuos) un ligero mejor valor de H para NSGAI que para CMOEA20-80&TOPSIS aunque las diferencias son ligeramente menores que las obtenidas al comparar CMOEA&TOPSIS y NSGAI.

6.2.9 Conclusiones

Los resultados de comparar TOPSISGA con NSGAI muestran que: (1) TOPSISGA focaliza la búsqueda de soluciones en la región de interés, (2) TOPSISGA induce un preorden completo en las soluciones encontradas, (3) NSGAI explora un frente de soluciones lo más amplio posible y (4) el porcentaje de soluciones obtenidas con TOPSISGA dominadas por las soluciones alcanzadas con NSGAI es menor (y por tanto mejor) para tamaños N pequeños de la población de soluciones no-dominadas, pero la diferencia desaparece al aumentar el tamaño N .

Las comparaciones llevadas a cabo con WP-TOPSISGA y R-NSGAI dejan ver que: (1) WP-TOPSISGA explora mejor que R-NSGAI, la región de interés (la R_d -PF), especialmente para valores del PF (en el espacio de los objetivos) escorados hacia los extremos del frente y alejados del frente de Pareto, (2) WP-TOPSISGA induce un preorden completo en las soluciones encontradas y (3) el porcentaje de soluciones obtenidas con WP-TOPSISGA dominadas por las soluciones alcanzadas con R-NSGAI es menor (y por tanto mejor) para tamaños N pequeños, de la población no-dominada y ubicaciones del PF hacia los extremos y alejados del frente de Pareto.

Tabla 6.7: Valores de la métrica relativa S (hipervolumen H), obtenidos al comparar los frentes completos alcanzados con CMOEA&TOPSIS y NSGAI.

Método	N=15	N=20	N=30	N=40	N=50	N=100
CMOEA&TOPSIS	1619	1615	1617	1620	1632	1631
NSGAI	1629	1630	1628	1630	1635	1630

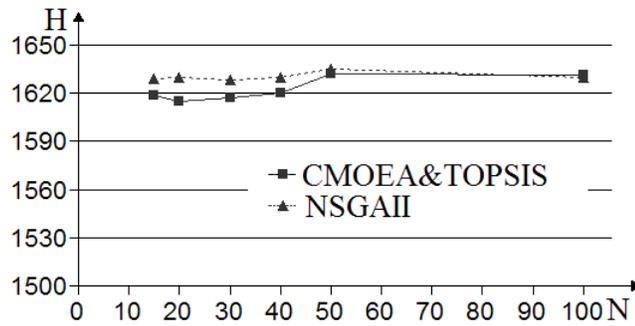


Figura 6.25: Visualización gráfica de la tabla 6.7.

Tabla 6.8: Valores porcentuales de la métrica relativa C, obtenidos al comparar los frentes completos alcanzados con CMOEA20-80&TOPSIS (frente B) y NSGAI (frente A).

	N=10	N=15	N=20	N=30	N=40	N=50	N=100
C(A,B)	16	23.1	32	35.29	38.04	37.92	37.68
C(B,A)	20	19.3	28.15	28.96	37.73	38.1	39.96

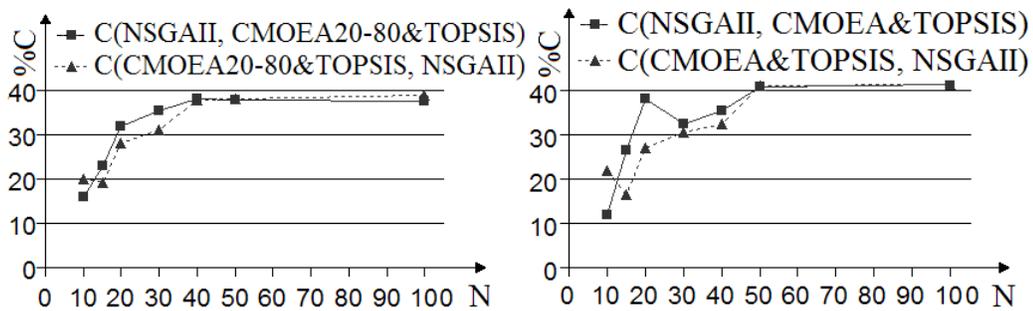


Figura 6.26: Visualización gráfica de las tablas 6.8 (izquierda) y 6.5 (derecha).

Tabla 6.9: Valores porcentuales de la métrica relativa C, obtenidos al comparar los frentes (RP) alcanzados con CMOEA20-80&TOPSIS (frente B) y NSGAI (frente A).

	N=10	N=15	N=20	N=30	N=40	N=50	N=100
C(A,B)	0	8	0	4	25.7	42.1	42.3
C(B,A)	100	88.32	95	59.6	68.6	41.9	42.5

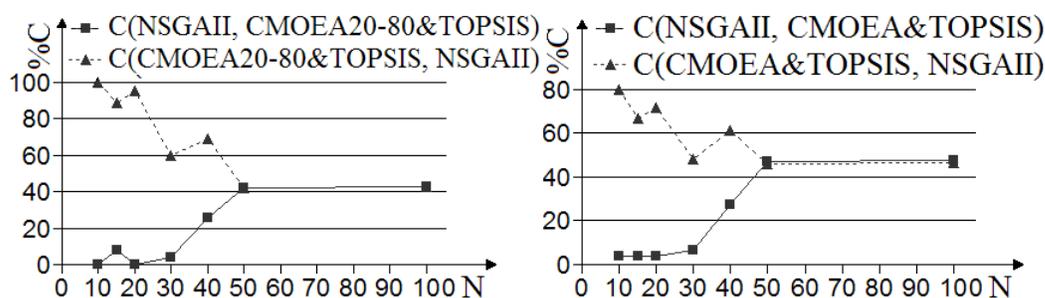


Figura 6.27: Visualización gráfica de las tablas 6.9 (izquierda) y 6.6 (derecha).

Tabla 6.10: Valores de la métrica relativa S (hipervolumen H), obtenidos al comparar los frentes completos alcanzados con CMOEA20-80&TOPSIS y NSGAI.

Método	N=15	N=20	N=30	N=40	N=50	N=100
CMOEA20-80&TOPSIS	1621	1619	1620	1622	1632	1631
NSGAI	1629	1630	1628	1630	1635	1630

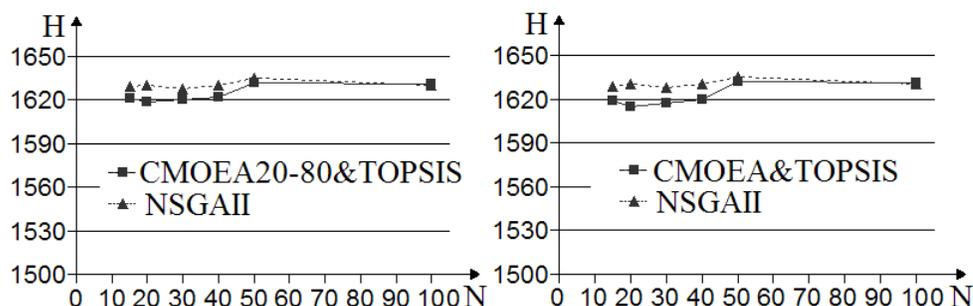


Figura 6.28: Visualización gráfica de las tablas 6.10 (izquierda) y 6.7 (derecha).

Los resultados de comparar CMOEA&TOPSIS y TOPSISGA, denotan un mejor porcentaje de la métrica C para CMOEA&TOPSIS que para TOPSISGA cuando el tamaño de la población de soluciones no-dominada es reducido ($N=15$).

La comparación entre CMOEA&TOPSIS y NSGAI (comparación completa del frente de soluciones obtenido), muestra una mejor evolución de CMOEA&TOPSIS con tamaños N pequeños de la población no-dominada. La situación se invierte (mejor comportamiento de NSGAI), para tamaños intermedios de N . Al incrementar N , los comportamientos de ambos algoritmos son similares. Sin embargo, cuando se compara la RP, CMOEA&TOPSIS es siempre para valores pequeños e intermedios mucho mejor que NSGAI. Las figuras 6.29 y 6.30 ayudan a interpretar los resultados de CMOEA&TOPSIS y NSGAI. En la región de precisión CMOEA&TOPSIS se comporta mucho mejor que NSGAI, sin embargo fuera de ella, a pesar de que CMOEA&TOPSIS obtiene un amplio y distribuido frente de soluciones, este es peor que el logrado por NSGAI. En cuanto a la comparativa con la métrica S o hipervolumen H, los resultados con tamaños hasta $N=50$ de la población no-dominada, son ligeramente mejores para NSGAI que para CMOEA&TOPSIS. Y es que, como muestran las figuras 6.29 y 6.30, la extensión del frente alcanzado fuera de la RP, es mayor que la obtenida en la RP, y es justamente fuera de la RP donde NSGAI se acerca más al frente real de Pareto que CMOEA&TOPSIS.

En las comparaciones (con las métricas C y S, y frentes de soluciones completos) hechas entre CMOEA20-80&TOPSIS y NSGAI, se observa que las diferencias a favor de NSGAI son algo menores que las obtenidas al comparar CMOEA&TOPSIS y NSGAI. Por otra parte, cuando se comparan los frentes de soluciones pertenecientes a la RP (con métrica C), el comportamiento de CMOEA20-80&TOPSIS comparado con NSGAI es mejor que el de CMOEA&TOPSIS confrontado a NSGAI. La interpretación de estos resultados es que CMOEA20-80&TOPSIS concentra algunas menos (máximo un 20% de N con CMOEA20-80&TOPSIS, mientras que un tamaño fijo e igual a 10 con CMOEA&TOPSIS) soluciones (aunque mejores) en la RP que CMOEA&TOPSIS (luego CMOEA20-80&TOPSIS es mucho mejor que NSGAI en la RP); de otra parte y fuera de la RP, CMOEA20-80&TOPSIS dispone de más soluciones que CMOEA&TOPSIS, por lo que los comportamientos de CMOEA20-80&TOPSIS y NSGAI son similares.

Es importante notar, que los métodos (TOPSISGA, WP-TOPSISGA, NSGAI, R-NSGAI, CMOEA&TOPSIS y CMOEA20-80&TOPSIS) consiguen con el problema de la Mochila Multiobjetivo (121 soluciones óptimas) presentado al principio del capítulo, un frente no-dominado (frente 1) de entre 40 y 50 soluciones, lo que explica los resultados similares que se obtienen con las métricas C y S para tamaños N de la población no-dominada alrededor o superior a 50 individuos (ya que los operadores

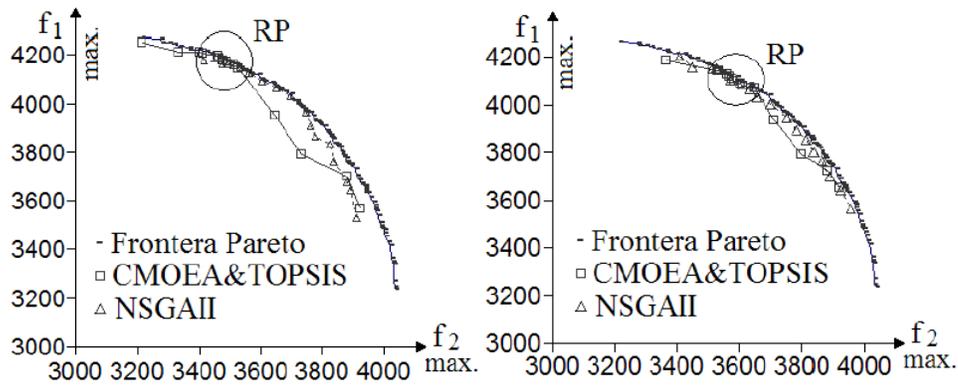


Figura 6.29: Gráfica para interpretación de los resultados obtenidos en la comparación de CMOEA&TOPSIS con NSGAII: $N=15$, $w_1=0.9$ $w_2=0.1$ (izquierda) y $w_1=0.75$ $w_2=0.25$ (derecha).

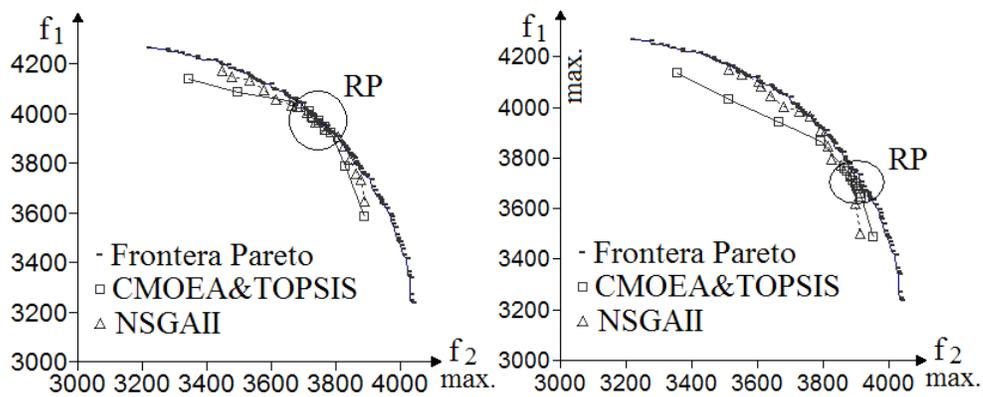


Figura 6.30: Gráfica para interpretación de los resultados obtenidos en la comparación de CMOEA&TOPSIS con NSGAII: $N=15$, $w_1=0.5$ $w_2=0.5$ (izquierda) y $w_1=0.1$ $w_2=0.9$ (derecha).

similarity de TOPSISGA o WP-TOPSISGA, crowding distance de NSGAI, preference operator de R-NSGAI y similarity+crowding distance de CMOEA&TOPSIS o CMOEA20-80&TOPSIS, están fuertemente limitados).

6.3 Problema de Optimización Multiobjetivo en Diseño de Sistemas de Seguridad

6.3.1 Diseño Óptimo de Sistemas de Seguridad

La función de un sistema de seguridad es actuar cuando ciertos eventos se producen, evitando averías o situaciones dañinas cuya aparición podría dar lugar a hechos graves. El diseño del sistema debe en lo posible asegurar el éxito de la misión y para ello el diseñador intenta encontrar entre diferentes alternativas, la realización física más conveniente del sistema, decidiendo el mejor compromiso tanto en la selección de dispositivos a instalar, como en la localización física de los componentes y, en algunas ocasiones en los intervalos temporales entre mantenimientos preventivos.

Frecuentemente, en problemas ODSS se utilizan como medidas de la validez del sistema la no-disponibilidad (probabilidad de que el sistema falle cuando se le demande cumplir su misión) y el coste (suma de los costos de los dispositivos elegidos para el sistema). Además, la tarea de la optimización está en la práctica restringida debido a las limitaciones de los recursos físicos y económicos.

En ODSS automatizados, un software ensaya diferentes dispositivos y configuraciones (serie, paralelo o mixtas) obteniendo el diseño óptimo del sistema considerando las restricciones del sistema. Este tipo de software posee dos componentes fundamentales: el modelo del sistema y el método de optimización.

El modelo del sistema

La metodología más difundida para modelar sistemas de seguridad es el análisis de árboles de fallos. Un árbol de fallos, consiste en un diagrama booleano, compuesto fundamentalmente por puertas lógicas AND y OR que modelan la lógica de fallos del sistema. La figura 6.31 tomada de [89], muestra un árbol de fallos simplificado (donde V significa válvula y B, bomba) del SCIS de una central nuclear representado en la figura 6.32.

Existen muchos métodos para la evaluación cuantitativa de los árboles de fallos, en esta tesis se utilizará un método eficiente denominado método de los pesos [78][87], que proporciona límites superior e inferior exactos de la no-disponibilidad del sistema para un error preespecificado máximo.

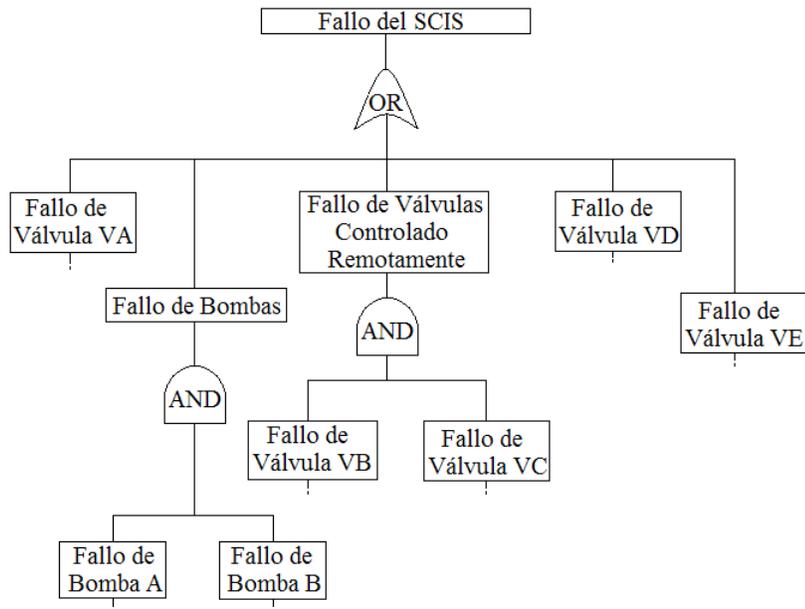


Figura 6.31: Arbol simplificado de fallos del SCIS de una central nuclear.

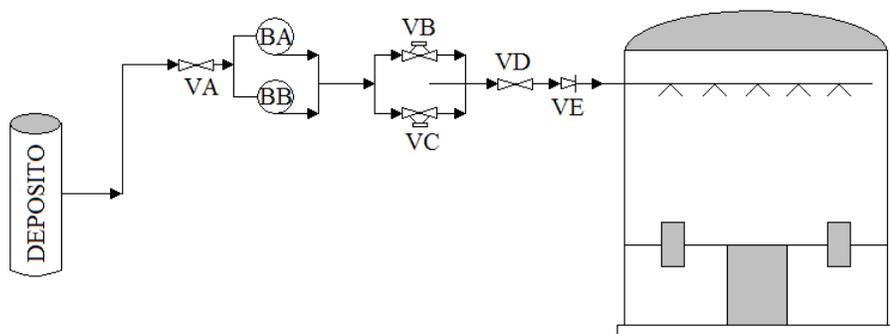


Figura 6.32: Diagrama simplificado del SCIS de una central nuclear.

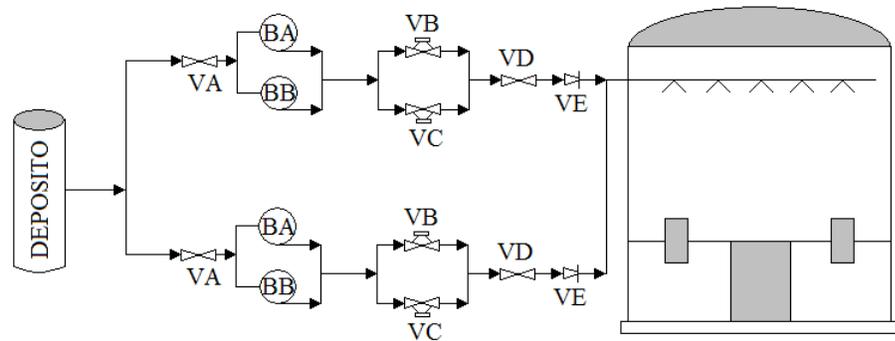


Figura 6.33: Diagrama del SCIS de una central nuclear utilizado como aplicación práctica.

El método de optimización

Los métodos más utilizados en la optimización de problemas de ODSS, son los algoritmos evolutivos [82][85][92][98]. Una consideración importante previa al uso del algoritmo evolutivo es la codificación de las soluciones candidatas. Código binario, entero y real puede ser usado, si bien el código binario es el más utilizado [3] y el que hemos adoptado para el problema utilizado como aplicación práctica de esta tesis, el cual se describe en la siguiente sección. No obstante, ciertos autores [150], han observado problemas de convergencia en ODSS a causa de la diferente naturaleza de las variables implicadas (variables binarias para las alternativas de diseño y variables reales en las estrategias de mantenimiento).

6.3.2 Aplicación práctica: Sistema de Contención por Inyección en Spray

El SCIS de una central nuclear para generación eléctrica, es un sistema cuya misión consiste en inyectar agua borada en el interior de la contención del edificio del reactor cuando se ha producido un accidente del tipo Pérdida de Refrigerante. Mediante aspersores situados en la parte superior del edificio, se procede a liberar el agua borada la cual arrastra la contaminación radiactiva conduciéndola a un depósito seguro para su posterior tratamiento. El sistema pasa por tanto su vida útil en espera de intervenir, siendo necesario asegurar su funcionamiento (disponibilidad) en el caso remoto de ser necesaria su intervención. En la figura 6.33 se puede apreciar la configuración general del sistema (se trata del mismo modelo del sistema de la figura 6.32, al cual se ha duplicado la línea por razones de seguridad). La optimización que se pretende, consiste en encontrar las diferentes combinaciones de componentes técnicos existentes

Tabla 6.11: Valores de parámetros del problema SCIS.

	VA, VD	VB, VC	VE	PA, PB
Modelo 1	P=2.9E-3 C=50	P=3.0E-3 C=65	O=5.0E-4 C=37	P=3.5E-3 C=90
Modelo 2	P=8.7E-3 C=35	P=1.0E-3 C=70	O=6.0E-4 C=35	P=3.8E-3 C=85
Modelo 3	P=4.0E-4 C=60	—	—	—

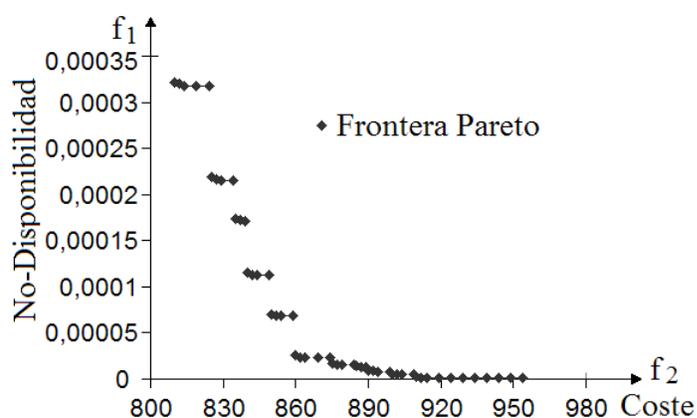


Figura 6.34: Frontera de Pareto del problema de SCIS.

en el mercado, que minimicen simultáneamente la no-disponibilidad del sistema y su coste. Dado que ambos objetivos están en conflicto, es evidente que el problema es de naturaleza multiobjetivo.

6.3.3 Datos adoptados

Los datos adoptados han sido los de Greiner [89]. En la tabla 6.11, se indican los valores de la no-disponibilidad (P) y coste (C) de los diferentes dispositivos del sistema que pueden encontrarse en el mercado. Tres modelos son posibles para las válvulas A y D, para el resto de válvulas y bombas sólo son posibles dos modelos. La no-disponibilidad y coste están restringidos a los valores $[0.0, 1.0]$ y $[0, 960]$ respectivamente.

El tamaño del espacio de búsqueda con codificación binaria simple de 32 bits es 4.294.967.296 soluciones. El número de soluciones factibles es de 82944. En cuanto al conjunto de soluciones óptimas de Pareto, este es 178 soluciones en el espacio de decisión y 51 soluciones en el espacio de los objetivos, los valores de estas últimas se dan en la tabla 6.12 y se muestran gráficamente en la figura 6.34.

Tabla 6.12: Valores soluciones del problema de SCIS.

no-disponibilidad	coste	no-disponibilidad	coste	no-disponibilidad	coste
0.0003217533	810	0.0000685586	852	0.0000083839	894
0.0003199908	812	0.0000681832	854	0.0000083462	899
0.0003182379	814	0.0000680768	859	0.0000056120	900
0.0003181330	819	0.0000255526	860	0.0000052204	902
0.0003180280	824	0.0000237605	862	0.0000050886	904
0.0002186917	825	0.0000236305	864	0.0000050657	909
0.0002169189	827	0.0000235239	869	0.0000020593	910
0.0002157307	829	0.0000234884	874	0.0000019171	912
0.0002156252	834	0.0000173781	875	0.0000017850	914
0.0001742693	835	0.0000161601	877	0.0000017771	919
0.0001724920	837	0.0000160294	879	0.0000017692	924
0.0001715472	839	0.0000159571	884	0.0000017666	929
0.0001150281	840	0.0000138547	885	0.0000017640	934
0.0001132449	842	0.0000128842	887	0.0000017625	939
0.0001126247	844	0.0000127531	889	0.0000017610	944
0.0001125186	849	0.0000091560	890	0.0000017596	949
0.0000703462	850	0.0000085154	892	0.0000017583	954

6.3.4 Configuración de parámetros

En todos los experimentos se utilizó un tamaño de la población R de 200 individuos, una probabilidad de cruce de 0.8, una probabilidad de mutación de 0.01, un valor $p=2$ para la métrica p en la ecuación (1.10.1) y el número máximo de generaciones (G) fue de 100. El tamaño N del fichero de soluciones no-dominadas, fue cambiado progresivamente según las secuencias $N=10,15,20,30,40,50,100$.

6.3.5 Métricas

Las métricas utilizadas para la evaluación y comparación de resultados fueron la métrica relativa $C(A,B)$ [222] para medir el porcentaje (media después de diez ensayos) de soluciones de B que son dominadas por al menos una solución de A (menor valor de $C(A,B)$, mejor para B y peor para A) (ver la sección 4.5.1.) y la métrica S [222] (mayor valor de S , mejor) para medir el tamaño (media después de diez ensayos) de la región dominada por el frente de Pareto y limitada por un punto de referencia (se adoptó el punto de coordenadas 1.00, 960.00) (ver la sección 4.5.2).

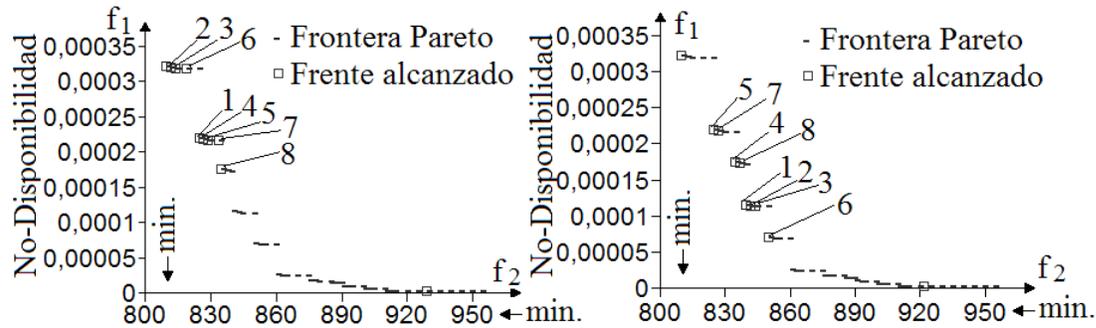


Figura 6.35: Frente óptimo y preorden logrados por TOPSISGA: $N=10$, $w_1=0.001$, $w_2=0.999$ (izquierda) y $w_1=0.005$, $w_2=0.995$ (derecha) con el problema de SCIS.

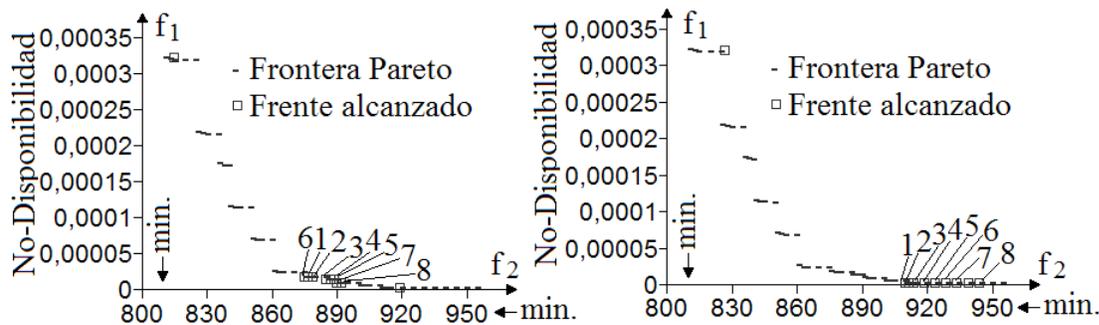


Figura 6.36: Frente óptimo y preorden logrados por TOPSISGA: $N=10$, $w_1=w_2=0.5$ (izquierda) y $w_1=0.99$, $w_2=0.01$ (derecha) con el problema de SCIS.

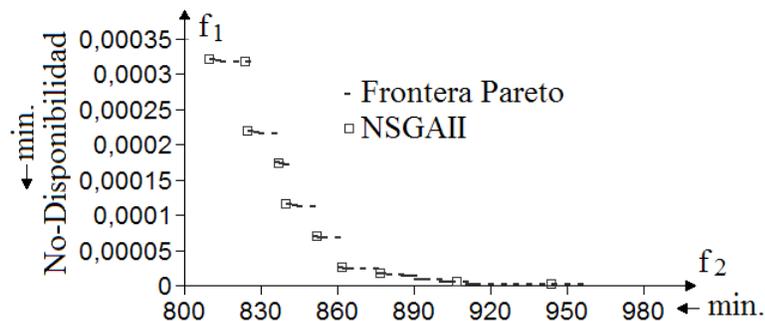


Figura 6.37: Frente alcanzado por NSGAII con el problema de SCIS.

Tabla 6.13: Valores en tanto por ciento de la métrica relativa $C(A,B)$, obtenidos al comparar los frentes alcanzados con TOPSISGA (frente B) y NSGAI (frente A) con el problema de SCIS.

	N=10	N=15	N=20	N=30	N=40	N=50	N=100
TOPSISGA	6	2.67	5	4.1	4.02	4.21	4.9
NSGAI	20	13.32	12	4	3.54	3.98	4.40

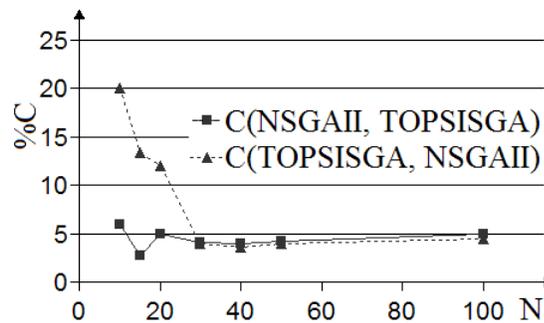


Figura 6.38: Vista gráfica de los valores de la tabla 6.13

6.3.6 Análisis de resultados con TOPSISGA

Las figuras 6.35 y 6.36 muestran los resultados obtenidos con TOPSISGA (para $N=10$) para diferentes valores de los pesos w_1 y w_2 sugeridos por el decisor. Las numeraciones corresponden al preorden completo inducido acorde al método TOPSIS. Se observa a partir de ambas figuras, como la región de exploración y el preorden final de las soluciones, cambian al variar los valores de los pesos. La figura 6.37 muestra el frente de soluciones alcanzado con NSGAI para un tamaño $N=10$ individuos de la población no-dominada. De las figuras, se observa que TOPSISGA explora una particular región de la frontera eficiente, mientras que NSGAI explora un amplio y bien distribuido conjunto eficiente de soluciones.

Comparativa TOPSISGA y NSGAI: métrica C

La tabla 6.13 muestra los valores obtenidos al comparar los frentes completos alcanzados por TOPSISGA y NSGAI con la métrica relativa C . Se observa que para tamaños $N=10,15,20$ pequeños de la población no-dominada, la medida $C(A,B)$ es menor (menos mejor) para TOPSISGA que para NSGAI, pero al incrementar el tamaño N las diferencias desaparecen. La figura 6.38 visualiza y ayuda a mejor interpretar los valores de la tabla 6.13.

Tabla 6.14: Valores en tanto por ciento de la métrica relativa C , obtenidos al comparar los frentes alcanzados con WP-TOPSISGA (frente B) y R-NSGAI (frente A).

	N=10	N=15
$(f_1=2.386E-04, f_2=837)$		
C(A,B)	0	0
C(B,A)	0	0
$(f_1=6.979E-06, f_2=914)$		
C(A,B)	0	0
C(B,A)	0	0

6.3.7 Análisis de resultados con WP-TOPSISGA

Las figuras 6.39 y 6.40 (para mayor claridad se ha hecho un zoom entre 0 y 0.00005 de la no-disponibilidad), muestran los frentes alcanzados por WP-TOPSISGA y R-NSGAI para tamaños de la población no-dominada $N=10$ individuos y valores del PF ($f_1=0.0002383$, $f_2=837$), ($f_1=0.000006979$, $f_2=914$) respectivamente. Dada la forma del frente real de Pareto no se observan prácticamente diferencias entre los conjuntos de soluciones no-dominadas alcanzados (en la R_d -PF) por ambos algoritmos. Las numeraciones corresponden al preorden completo inducido (con WP-TOPSISGA) según TOPSIS.

Comparativa WP-TOPSISGA y R-NSGAI: métrica C

No se aprecian diferencias significativas cuando se comparan los frentes alcanzados (en la R_d -PF) por WP-TOPSISGA y R-NSGAI en base a la métrica relativa C (véase la tabla 6.14). En efecto, la forma del frente real de Pareto del problema SCSI tratado aquí, hace que los comportamientos de WP-TOPSISGA y R-NSGAI sean similares.

6.3.8 Análisis de resultados con CMOEA&TOPSIS

Las figuras 6.41, 6.42, 6.43, 6.44, 6.45 y 6.46 muestran los frentes óptimos de soluciones alcanzados por CMOEA&TOPSIS y el preorden completo en base a la similaridad [45] inducido en dichas soluciones. En todas ellas se remarcan dos zonas diferenciadas del frente, una zona con una elevada densidad de soluciones (se trata de la RP con 10 SP) y la otra (fuera de la RP), que esboza ampliamente el resto del frente óptimo. Remarcar, como la RP explorada difiere en función de los pesos acordados por el decisor.

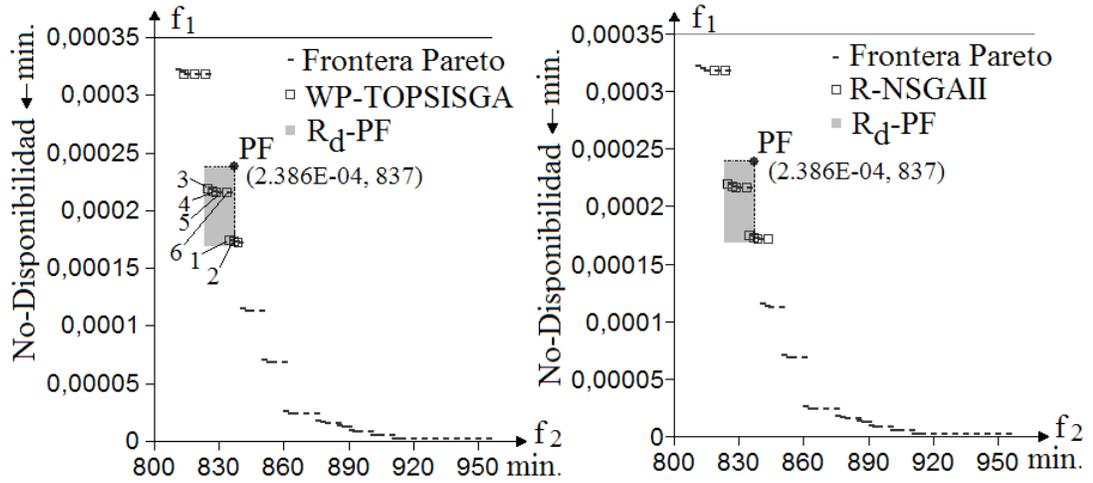


Figura 6.39: Frente alcanzado (N=10) con WP-TOPSISGA con el problema de SCIS.

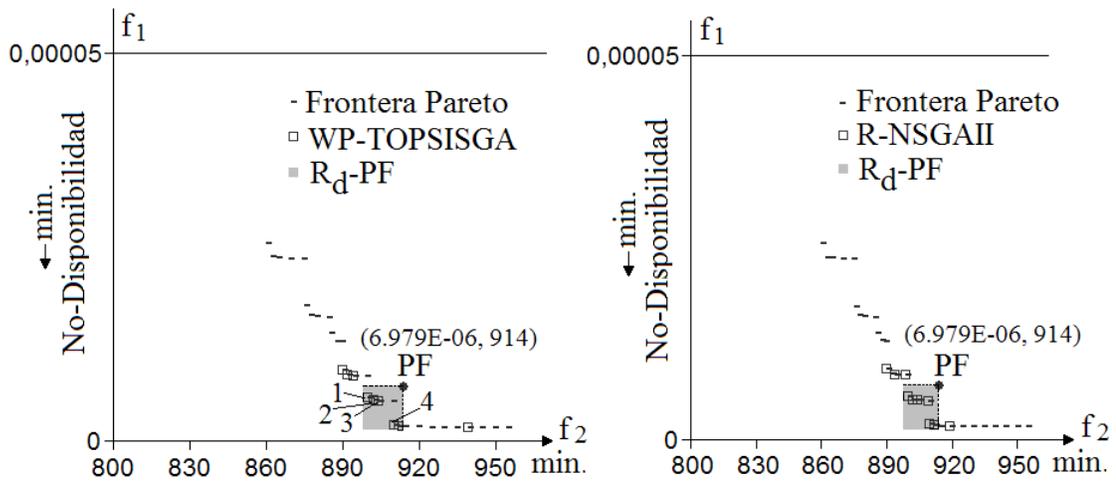


Figura 6.40: Frente alcanzado (N=10) con WP-TOPSISGA con el problema de SCIS.

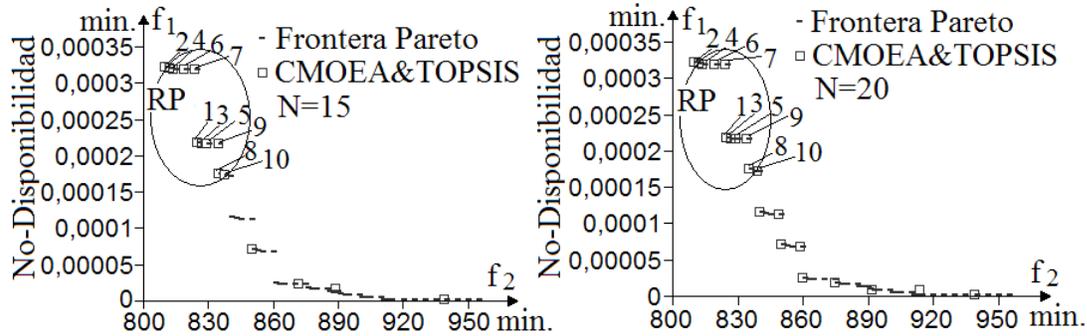


Figura 6.41: Problema de SCIS. Frente óptimo y ordenación TOPSIS logrados con CMOEA&TOPSIS: $SP=10$, $w_1=0.001$ $w_2=0.999$ y $N=15$ (izquierda), $N=20$ (derecha).

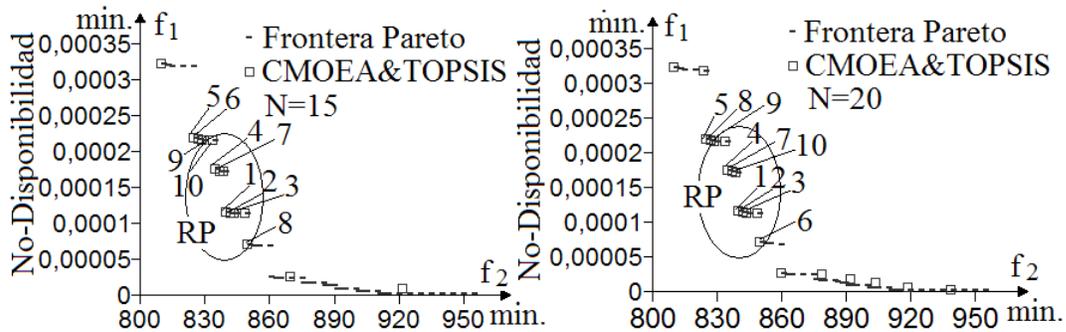


Figura 6.42: Problema de SCIS. Frente óptimo y ordenación TOPSIS logrados con CMOEA&TOPSIS: $SP=10$, $w_1=0.005$ $w_2=0.995$ y $N=15$ (izquierda), $N=20$ (derecha).

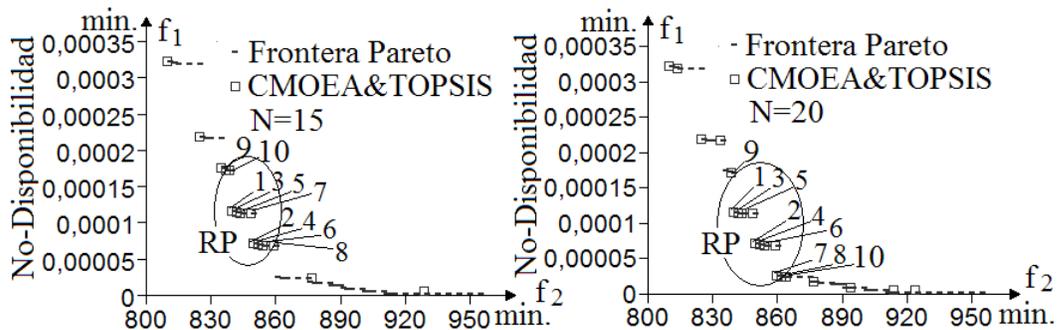


Figura 6.43: Problema de SCIS. Frente óptimo y ordenación TOPSIS logrados con CMOEA&TOPSIS: $SP=10$, $w_1=0.01$ $w_2=0.99$ y $N=15$ (izquierda), $N=20$ (derecha).

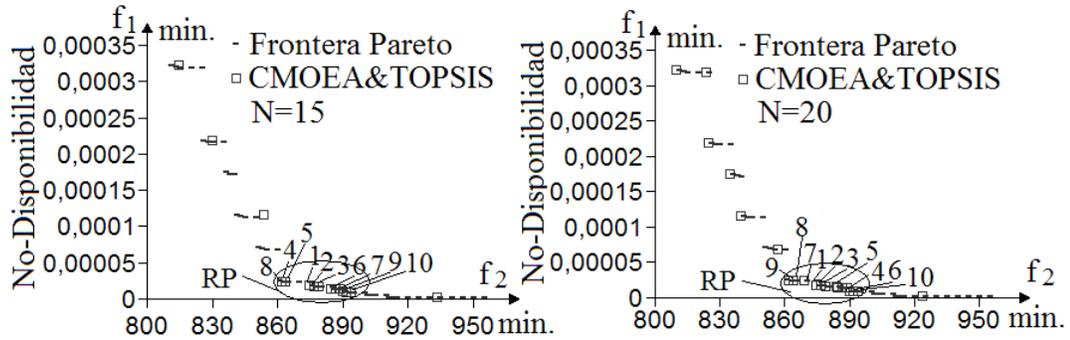


Figura 6.44: Problema de SCIS. Frente óptimo y ordenación TOPSIS logrados con CMOEA&TOPSIS: SP=10, $w_1=w_2=0.5$ y N=15 (izquierda), N=20 (derecha).

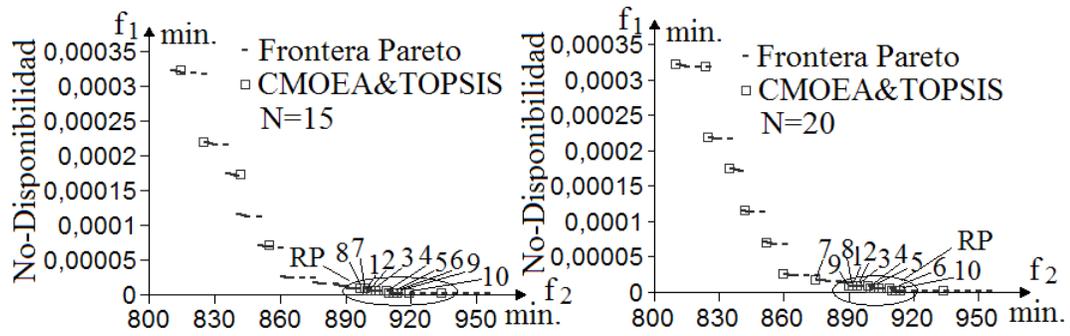


Figura 6.45: Problema de SCIS. Frente óptimo y ordenación TOPSIS logrados con CMOEA&TOPSIS: SP=10, $w_1=0.9$ $w_2=0.1$ y N=15 (izquierda), N=20 (derecha).

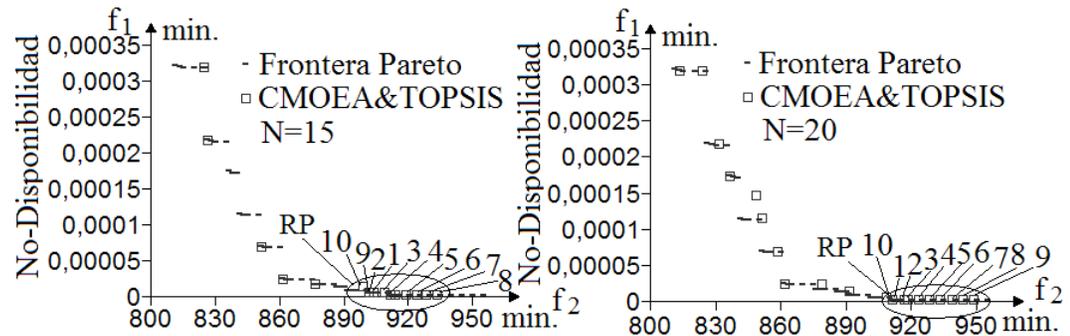


Figura 6.46: Problema de SCIS. Frente óptimo y ordenación TOPSIS logrados con CMOEA&TOPSIS: SP=10, $w_1=0.99$ $w_2=0.01$ y N=15 (izquierda), N=20 (derecha).

Comparativa CMOEA&TOPSIS y TOPSISGA: métrica C

La tabla 6.15 muestra los valores obtenidos al comparar los frentes completos alcanzados por CMOEA&TOPSIS y TOPSISGA con la métrica relativa C. Se aprecia una cierta ventaja de CMOEA&TOPSIS para un tamaño $N=15$ del fichero de soluciones no-dominadas, para el resto de tamaños de N , los resultados son similares para ambos algoritmos. La figura 6.47 visualiza los resultados de la tabla 6.15.

Comparativa CMOEA&TOPSIS y NSGAII: métricas C y S

Los resultados de comparar (mediante la métrica relativa C) los frentes de soluciones completos y la RP, obtenidos con CMOEA&TOPSIS y NSGAII se muestran en las tablas 6.16, 6.17 y figuras 6.48, 6.49. Cuando se comparan los frentes completos, el valor porcentual de C, es mejor para CMOEA&TOPSIS para tamaños N de la población no-dominada comprendidos entre los valores 10 y 20. Sin embargo, para tamaños N entre 20 y 40, los mejores valores de C son para NSGAII. Con mayores tamaños de N , las diferencias desaparecen. Cuando se comparan los frentes de soluciones pertenecientes a la RP, los valores de C, son notablemente mejores para CMOEA&TOPSIS hasta tamaños $N=40$, al incrementar N los valores de C se igualan para ambos algoritmos.

La tabla 6.18 y figura 6.50 muestran los resultados alcanzados con la métrica S (hipervolumen H). Para tamaños de N menores a 50 individuos, el valor de la métrica S es mejor para NSGAII que para CMOEA&TOPSIS, para mayores tamaños de N , los resultados se igualan.

6.3.9 Análisis de resultados con CMOEA20-80&TOPSIS

Comparativa CMOEA20-80&TOPSIS y NSGAII: métricas C y S

La tabla 6.19 muestra los valores obtenidos al comparar los frentes completos alcanzados con CMOEA20-80&TOPSIS y NSGAII con la métrica relativa C. La figura 6.51 izquierda (representa gráficamente los resultados de la tabla 6.19) muestra comparándola con la figura 6.51 derecha (resultados visuales de la tabla 6.16 de comparar CMOEA&TOPSIS y NSGAII), que los valores de la métrica C son mejores para CMOEA20-80&TOPSIS con N comprendido entre 10 y 20 individuos mientras que para tamaños N con valores entre 20 y 40 individuos el valor de la métrica C es mejor para NSGAII, aunque las diferencias se han reducido.

Cuando se comparan los frentes obtenidos con CMOEA20-80&TOPSIS y NSGAII correspondientes a la RP, los resultados con la métrica C, son algo mejores para CMOEA20-80&TOPSIS comparados con NSGAII que los de la comparación entre CMOEA&TOPSIS y NSGAII, para valores N entre 10 y 30 individuos, pero

Tabla 6.15: Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes completos alcanzados con CMOEA&TOPSIS (frente B) y TOPSISGA (frente A) con el problema SCIS.

	N=10	N=15	N=20	N=30	N=40	N=50	N=100
C(A,B)	6	3.9	5.02	6.85	6.5	6.54	6.32
C(B,A)	6.2	6.3	5.1	7.12	6.02	6.62	6.41

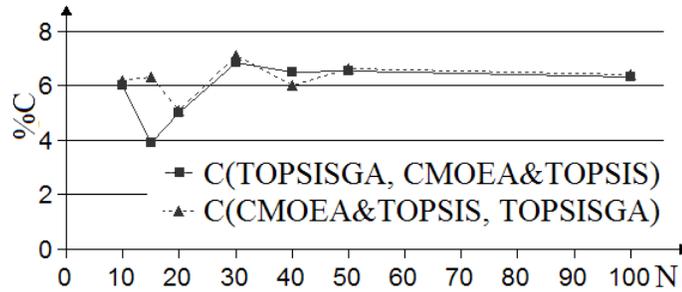


Figura 6.47: Visualización gráfica de la tabla 6.15.

Tabla 6.16: Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes completos alcanzados con CMOEA&TOPSIS (frente B) y NSGAI (frente A) con el problema SCIS.

	N=10	N=15	N=20	N=30	N=40	N=50	N=100
C(A,B)	2	1.33	10.02	15.32	5	6.32	6.26
C(B,A)	10	10.64	9.98	2.66	6.1	6.09	6.33

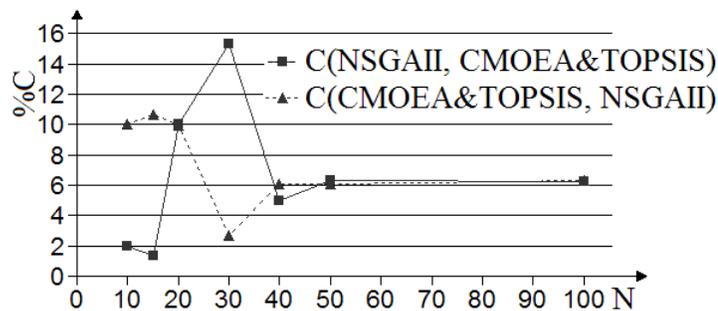


Figura 6.48: Visualización gráfica de la tabla 6.16.

Tabla 6.17: Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes (RP) alcanzados con CMOEA&TOPSIS (frente B) y NSGAI (frente A) con el problema SCIS.

	N=10	N=15	N=20	N=30	N=40	N=50	N=100
C(A,B)	0	2	0	0	0	0	0
C(B,A)	40	36.6	13	10	0	0	0

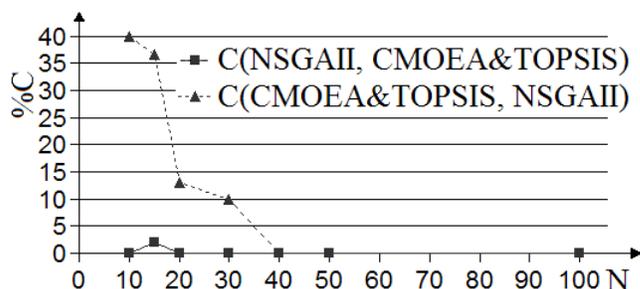


Figura 6.49: Visualización gráfica de la tabla 6.17.

Tabla 6.18: Valores de la métrica relativa S (hipervolumen H), obtenidos al comparar los frentes completos alcanzados con CMOEA&TOPSIS y NSGAI con el problema de SCIS.

Método	N=15	N=20	N=30	N=40	N=50	N=100
CMOEA&TOPSIS	140.58	143.184	143.58	146.98	149.98	149.98
NSGAI	149.98	149.98	149.51	149.98	149.98	149.98

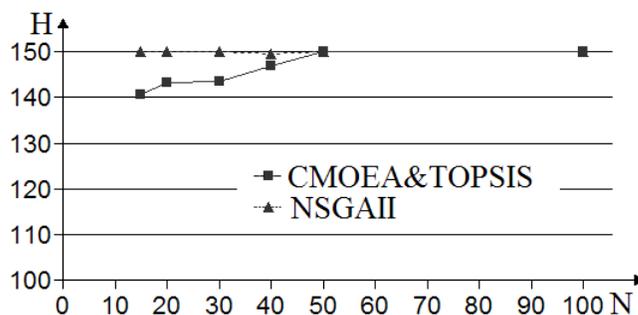


Figura 6.50: Visualización gráfica de la tabla 6.18.

Tabla 6.19: Valores en tanto por ciento de la métrica relativa C, obtenidos al comparar los frentes completos alcanzados con CMOEA20-80&TOPSIS (frente B) y NSGAI (frente A) con el problema de SCIS.

	N=10	N=15	N=20	N=30	N=40	N=50	N=100
C(A,B)	4	5.32	10.08	10.65	6.45	6.91	6.31
C(B,A)	12	10.32	10.2	4.10	6.18	6.23	6.33

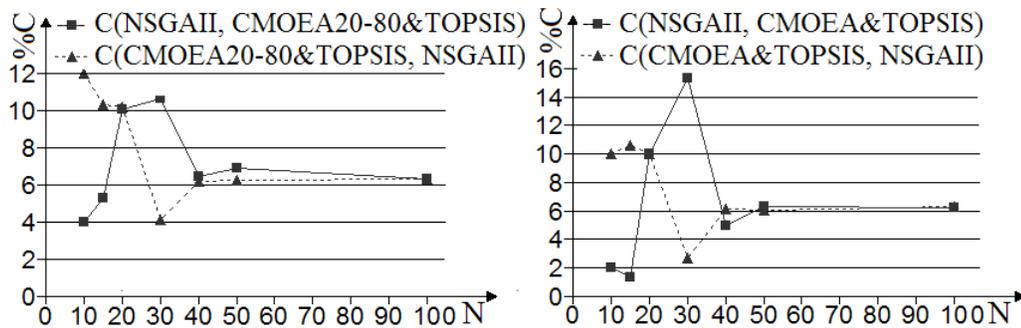


Figura 6.51: Visualización gráfica de la tabla 6.19.

desaparecen las diferencias (ver la tabla 6.20 y figuras 6.52 izquierda y 6.52 derecha obtenida de 6.17) cuando se incrementan los tamaños de N.

Para la métrica S o hipervolumen H (se comparan los frentes a partir de N=15 individuos), se observan mejores valores de H para NSGAI que para CMOEA20-80&TOPSIS, pero esta diferencias son menores que las obtenidas al comparar NSGAI con CMOEA&TOPSIS (ver la tabla 6.21 y figuras 6.53 izquierda y 6.53 derecha obtenida de la tabla 6.18).

6.3.10 Conclusiones

Cuando se comparan TOSISGA y NSGAI en el problema SCIS, se observa que TOPSISGA centra la exploración del frente óptimo, en aquella región que resulta de interés para un decisor, clasificando las soluciones encontradas según el método TOPSIS [101]. Por su parte, NSGAI explora el frente óptimo lo más ampliamente posible. En cuanto a la comparativa en base a la métrica C, los valores de ésta, son mejores para TOPSISGA cuando los tamaños N del fichero de soluciones no-dominadas son pequeños, desapareciendo las diferencias cuando se incrementa el tamaño N.

Los resultados que se obtuvieron (tanto visualmente, como en base a la métrica C) al comparar WP-TOPSISGA con R-NSGAI fueron muy similares, aunque hay

Tabla 6.20: Problema de SCIS. Porcentajes de la métrica C, de comparar los frentes (RP) alcanzados con CMOEA20-80&TOPSIS (frente B) y NSGAI (frente A).

	N=10	N=15	N=20	N=30	N=40	N=50	N=100
C(A,B)	0	5	2	0	0	0	0
C(B,A)	40	20	10	0	0	0	0

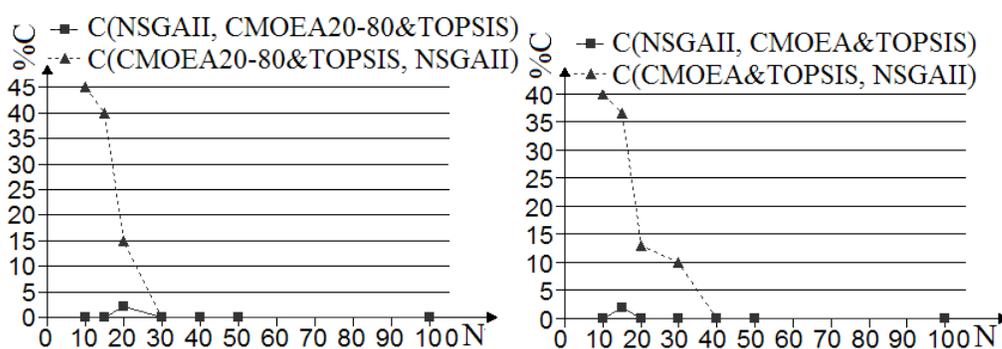


Figura 6.52: Visualización gráfica de la tabla 6.20.

Tabla 6.21: Valores de la métrica relativa S (hipervolumen H), obtenidos al comparar los frentes completos alcanzados con CMOEA20-80&TOPSIS y NSGAI con el problema de SCIS.

Método	N=15	N=20	N=30	N=40	N=50	N=100
CMOEA20-80&TOPSIS	144.32	144.86	146.58	148.98	149.98	149.98
NSGAI	149.98	149.98	149.51	149.98	149.98	149.98

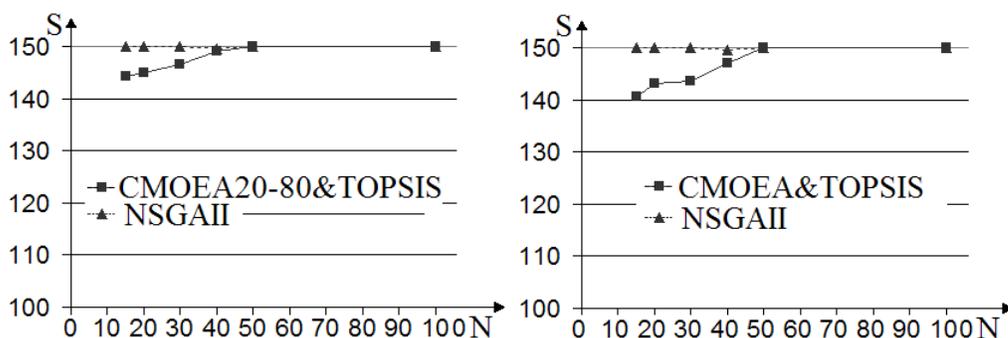


Figura 6.53: Visualización gráfica de la tabla 6.21.

que decir que la propuesta WP-TOPSISGA da un orden a las soluciones encontradas según la similaridad [45] a la distancia ideal.

Las comparaciones entre CMOEA&TOPSIS y TOPSISGA, denotan una pequeña diferencia del valor de la métrica C en favor de CMOEA&TOPSIS, sólo cuando se usan tamaños $N=15$ individuos, cualquier otro tamaño de N , conduce a comportamientos similares de ambos algoritmos.

Cuando se comparan los frentes completos obtenidos por CMOEA&TOPSIS y NSGAI, se observa (con la métrica C) un mejor comportamiento de CMOEA&TOPSIS para tamaños N entre 10 y 15 individuos, pero para tamaños de N comprendidos entre 20 y 40 individuos es NSGAI quien tiene ventaja. Para tamaños N igual o mayores a cuarenta, ambos algoritmos se comportan de forma similar. Es cuando se comparan los frentes alcanzados y pertenecientes a la RP, donde se aprecia notable ventaja de CMOEA&TOPSIS sobre NSGAI. De otra parte, los resultados de la métrica S, dan una pequeña ventaja a NSGAI sobre CMOEA&TOPSIS. La interpretación de estos resultados es similar a la que se hizo con los resultados obtenidos con el problema de la Mochila (sección 6.2.9): CMOEA&TOPSIS concentra y mejora ampliamente (en la Región de Precisión) los resultados obtenidos con NSGAI, pero fuera de ella, CMOEA&TOPSIS consigue un amplio y distribuido frente, pero con peor calidad de soluciones (más distantes del frente real de Pareto) que el logrado por NSGAI.

Las conclusiones extraídas de comparar CMOEA&TOPSIS con NSGAI se pueden extrapolar a la comparación hecha entre CMOEA20-80&TOPSIS y NSGAI, pero los resultados de CMOEA20-80&TOPSIS comparado con NSGAI son mejores que los de CMOEA&TOPSIS confrontado a NSGAI, tanto en la Región de Precisión como fuera de ella. Con CMOEA20-80&TOPSIS se concentran en la Región de Precisión, menos soluciones (máximo un 20% de N con CMOEA20-80&TOPSIS mientras, que un tamaño fijo e igual a 10 con CMOEA&TOPSIS) de mejor calidad (darse cuenta que las ventajas de CMOEA20-80&TOPSIS y CMOEA&TOPSIS frente a NSGAI son siempre para tamaños N pequeños de la población no-dominada) y de otra parte, hay más soluciones fuera de la RP con CMOEA20-80&TOPSIS que con CMOEA&TOPSIS, lo que hace que CMOEA20-80&TOPSIS tienda a comportarse como NSGAI.

Por último, anotar que los operadores similarity de TOPSISGA o WP-TOPSISGA, similarity+crowding distance de CMOEA&TOPSIS o CMOEA20-80&TOPSIS, preference operator de R-NSGAI y crowding distance de NSGAI están grandemente limitados para tamaños N de la población no-dominada próxima al tamaño real del frente de Pareto (51 soluciones para el problema de SCIS), de ahí los resultados similares con todos los métodos para tamaños N cercanos o superiores a los 50 individuos.

Capítulo 7

Conclusiones y Líneas Futuras

7.1 Introducción

Se presentan en esta tesis, un conjunto de trabajos de investigación que giran en torno a tres puntos fundamentales:

- Los Algoritmos Evolutivos Multiobjetivo (AEMOs) *a priori* (el decisor manifiesta sus preferencias -por ejemplo especificando una función de utilidad sobre los diferentes objetivos- antes del inicio del proceso de búsqueda). Los AEMOs, a diferencia de otros métodos de optimización (que en su mayoría) operan sobre una única solución, manipulan una población de soluciones. Esta diferencia les provee de una gran pericia a la hora buscar múltiples soluciones Pareto óptimas, en problemas multiobjetivo NP-difíciles con espacios de soluciones ya sean continuos, no continuos, discretos, no convexos, etc. Además, el enfoque *a priori*, ha experimentado en los últimos años un gran crecimiento en publicaciones, congresos, investigaciones e incluso en aplicaciones que se hacen del mismo. A ello han contribuido recientes investigaciones [1][52][53][56] que muestran, las dificultades de resolver problemas con alta dimensionalidad en el número de objetivos (en inglés Many Objective Optimisation Problems). De otra parte, algunos autores como por ejemplo Branke y Deb [15] y Branke et al. [16], sugieren que si un decisor tiene alguna idea acerca de que soluciones pueden ser preferidas, este conocimiento debe ser utilizado en focalizar la búsqueda en las regiones más interesantes del frente de Pareto.
- Los Problemas de Optimización Multiobjetivo (POM) definidos en el dominio del Discreto y calificados de NP-difíciles (la resolución exacta no es posible en un tiempo proporcional a N^n , siendo N el número de parámetros desconocidos del problema, y n una constante entera). Varias razones nos han llevado a porqué escogerlos. La primera es que en el paradigma Multiobjetivo, son

mucho menos utilizados y por tanto menos conocidos que los problemas continuos. La segunda razón es la existencia de una gran cantidad de problemas reales y situaciones prácticas que se modelan en el discreto. Además, y tercera razón, como se señala en la sección 4.1 de esta tesis, cuando se resuelven con AEMOs problemas discretos, la idea de proporcionar a un decisor un representativo conjunto de soluciones en términos del espacio de los objetivos, para que elija una según sus preferencias, puede no resultar lo más apropiado. Por ejemplo, la elección de un tamaño para el fichero de soluciones no-dominadas del AEMO, menor al número (desconocido) de soluciones del problema, obliga a perder soluciones potencialmente atractivas para el decisor [135]. Además, se ha encontrado que procedimientos de reducción de soluciones (por ejemplo el crowding distance o distancia de agrupamiento de NSGAI [51][55]) puede, en zonas muy pobladas, eliminar soluciones Pareto óptimas y conservar otras que aunque son no-dominadas en la población en curso, no son Pareto óptimas [158]. Como problema test, se utilizó el problema (NP-difícil) bi-objetivo ZDT5 de Deb et al. [220] formulado en variables binarias. Como problemas de aplicaciones se utilizaron el problema de la Mochila Multiobjetivo en variables binarias (0-1MOKP siglas en inglés de Multiobjective Knapsack Problem), calificado de NP-difícil y utilizado ampliamente en la resolución de problemas multiobjetivo en la comunidad de algoritmos evolutivos [105][107][135][174][219] y un problema multiobjetivo de Optimización en Diseño de Sistemas de Seguridad (ODSS), en particular se usó el Sistema de Contención por Inyección en Spray (SCIS) de una central nuclear de generación eléctrica, también de naturaleza NP-difícil y ampliamente estudiado en [78][89][91].

- La Ayuda Multicriterio a la Decisión. Esta disciplina persigue ayudar a un decisor, a elegir entre varias acciones (soluciones) según varios criterios (objetivos). Sus métodos, además de afrontar las problemáticas de Roy [161][163] (ver también la sección 1.6), no son difíciles de incorporar en un AEMO, razones que nos han hecho interesarnos por ellos. En concreto, nos ha parecido interesante adoptar el método (muy conocido en la Ayuda Multicriterio a la Decisión) TOPSIS [101] (Technique for Order Preference by Similarity to an Ideal Solution). El método se cimienta en el axioma de Zeleny [218]: *es racional elegir una acción lo más próxima a la ideal o lo más alejada de la anti-ideal* y en la situación más general, establece que se debe elegir aquella acción o solución con menor distancia a la ideal y mayor distancia a la acción o solución anti-ideal. El método TOPSIS, del cual no tenemos conocimiento que haya sido utilizado en el contexto de los algoritmos evolutivos multiobjetivo, lo hemos combinado con un potente AEMO de segunda generación, en particular hemos utilizado

NSGAI [51][55], aunque otra metodología evolutiva multiobjetivo puede ser usada.

7.2 Resumen de las Propuestas

A continuación, se resumen y destacan las características más importantes de los algoritmos evolutivos multiobjetivo propuestos en esta tesis.

La primera propuesta TOPSISGA, desarrollada para trabajar con problemas discretos del mundo real, afronta aquellas situaciones en las que si el frente óptimo de soluciones es muy numeroso y el tamaño de la población no-dominada está limitado, el AEMO no puede contener el conjunto completo de soluciones, obligando a que se pierdan soluciones potenciales de interés para el decisor. Un modo de superar esta dificultad, consiste en concentrar la exploración y búsqueda, promoviendo durante el proceso de búsqueda, aquellas soluciones situadas en la región de interés del decisor. En esta idea, el algoritmo propuesto combina las metodologías evolutivas multiobjetivo de segunda generación (se usa NSGAI [51][55]) con el método bien conocido de Toma de Decisiones Multicriterio TOPSIS [101]. El método guía la búsqueda hacia la región de interés del frente óptimo y clasifica (Problemática P_γ de Roy [161][163]) las soluciones encontradas, induciendo en ellas un preorden completo según la similaridad [45] a la solución ideal. En cuanto a las comparaciones realizadas entre TOPSISGA y NSGAI en base a la métrica relativa C [222], TOPSISGA parece tener un mejor comportamiento que NSGAI para tamaños pequeños de la población no-dominada (aunque es conveniente hacer balance entre el amplio frente producido por NSGAI con la identificación y exploración de la zona de interés realizada por TOPSISGA). Esta ventaja de TOPSISGA desaparece al incrementarse el tamaño de la población no-dominada, pues los operadores de reducción de soluciones (similarity de TOPSISGA o crowding distance de NSGAI) están muy limitados para tamaños N de la población no-dominada iguales o superiores al tamaño real del frente de Pareto.

El algoritmo WP-TOPSISGA que se presenta como segunda propuesta, está pensado para la resolución de problemas industriales, económicos u otros, caracterizados en el espacio de la decisión por un Punto de Funcionamiento (PF) [134][136], el cual identifica una solución estable y realizable de trabajo no necesariamente eficaz o Pareto óptima (también el PF se define como aquella solución que determina el estado actual del sistema). Estas definiciones del PF se proponen en el capítulo dos de la tesis, donde además se aportan dos axiomas de pertenencia del PF, al conjunto realizable de soluciones en el espacio de decisión y al conjunto realizable de soluciones en el espacio de los objetivos. Admitimos, que un decisor inmerso en la tarea de mejorar el PF, puede estar interesado en un pequeño número de soluciones tal que, localizadas en una cierta región del frente óptimo (la R_d -PF) [134] dominan el PF.

WP-TOPSISGA destaca por ser capaz de conducir la búsqueda de soluciones hacia la región de interés (la R_d -PF) y clasificar (afrentando por tanto la Problemática P_γ de Roy [161][163]) las soluciones alcanzadas, induciendo en ellas un preorden completo en base al método TOPSIS. El algoritmo, mezcla la segunda generación de AEMOs (se utiliza NSGAI), el método de Toma de Decisiones Multicriterio TOPSIS y el PF. Se destaca, que con el método TOPSIS, un decisor debe especificar una función de utilidad (pesos) sobre los diferentes objetivos, sin embargo el método propuesto, estima los pesos de forma automática por interpolación en cada iteración del algoritmo en base al valor del PF. Al contrastar WP-TOPSISGA con R-NSGAI [56] en base a la métrica relativa C , se extrae que los resultados comparativos, dependen considerablemente de la forma del frente óptimo del problema que se considere. Así por ejemplo, con el problema de Sistema de Contención por Inyección en Spray (SCIS) [78][89][91], no se observaron diferencias entre WP-TOPSISGA y R-NSGAI. Sin embargo, con los problemas ZDT5 de Deb et al. [220] y la Mochila Multiobjetivo [105][107][135][174][219], se encontró un mejor comportamiento de WP-TOPSISGA para posiciones escoradas del PF hacia los extremos de los frentes (extremo superior con el problema ZDT5 y ambos extremos con el problema de la Mochila Multiobjetivo). De cualquier forma, WP-TOPSISGA nunca tuvo un peor comportamiento que R-NSGAI.

Continuando con nuestros trabajos y como tercera propuesta, hemos desarrollado CMOEA&TOPSIS, el cual mejora TOPSISGA en dos aspectos: primero ofrece un mayor enriquecimiento del proceso de diversificación de soluciones en la población y segundo, afronta aquella situación en la que un decisor expresa qué zona del frente óptimo es de su interés, pero no está seguro de ello. El algoritmo, encuentra un amplio y bien distribuido frente óptimo de soluciones, pero enfatiza mediante una acentuada y constante densidad de soluciones, aquella región (que hemos denominado Región de Precisión RP y la cual debe ser definida mediante un valor del parámetro SP o número de Soluciones de Precisión deseadas por el decisor) del frente óptimo que resulta de interés para el decisor, clasificándola y resolviendo la Problemática P_γ de Roy [161][163] en base a la similaridad [45] a la solución ideal. En esta idea, se combinan en un único algoritmo, las buenas prestaciones de los mejores AEMOs de segunda generación (hemos utilizado el operador de agrupamiento -crowding- de NSGAI, aunque otros pueden ser usados) y las habilidades de los AEMOs a priori (se ha usado el operador similaridad -similarity- de TOPSISGA). Los resultados de comparar CMOEA&TOPSIS con TOPSISGA en base a la métrica relativa C , demuestran la importancia del proceso de diversificación de soluciones, pues en todos los problemas, los valores de la métrica C fueron mejores para CMOEA&TOPSIS. Al comparar CMOEA&TOPSIS con NSGAI, los resultados dan un valor mucho mejor de la métrica relativa C en la RP para CMOEA&TOPSIS que para NSGAI. Sin embargo

y pese a que CMOEA&TOPSIS obtiene fuera de la RP un amplio y bien distribuido frente de soluciones, los valores del hipervolumen S [222] y la métrica relativa C (fuera de la RP) son mejores para NSGAI que para CMOEA&TOPSIS. Estos resultados se explican, por las diferencias existentes tanto en número, como en disposición de las soluciones de la población no-dominada en CMOEA&TOPSIS y NSGAI. NSGAI distribuye homogéneamente las N soluciones de la población no-dominada. Sin embargo, CMOEA&TOPSIS concentra un número SP de soluciones en la RP (mayor que las que concentra NSGAI) y el comportamiento de CMOEA&TOPSIS en la RP es mejor que el de NSGAI, de otro lado y fuera de la RP, CMOEA&TOPSIS dispone de un menor número ($N-SP$) de soluciones a distribuir que NSGAI y es NSGAI quien obtiene mejores resultados.

Por último, con la propuesta CMOEA20-80&TOPSIS, se soluciona el inconveniente existente en CMOEA&TOPSIS, de tener que introducir el valor del parámetro “tamaño SP de la Región de Precisión”. Este valor, es ahora calculado de forma automática en cada iteración por el propio algoritmo, según la ley 20-80 de Pareto [149] (en cada iteración, el valor SP es el resultado de calcular el 20% del número de soluciones del frente que no cabe completamente en el fichero de soluciones no-dominadas). Los resultados finales, son mejores para CMOEA20-80&TOPSIS que los obtenidos con CMOEA&TOPSIS utilizando un tamaño (SP) fijo de 10 soluciones. En la Región de Precisión, el valor de la métrica C son aún mejores para CMOEA20-80&TOPSIS que los obtenidos en la comparación que se hizo entre CMOEA&TOPSIS y NSGAI (en los tres problemas que se utilizan en la tesis, el valor SP estimado en cada iteración del algoritmo según la regla de Pareto suele ser inferior a 10, y como se dijo en el resumen de la propuesta TOPSISGA, éste tiene un mejor comportamiento que NSGAI al reducir el tamaño de la población no-dominada). Los valores del hipervolumen S y de la métrica C (fuera de la Región de Precisión), siguieron siendo mejores para NSGAI, pero las diferencias se redujeron con respecto a las que se obtuvieron al comparar CMOEA&TOPSIS y NSGAI (ahora el número de soluciones que utiliza CMOEA20-80&TOPSIS fuera de la RP, es parecido al de NSGAI y los comportamientos de ambos algoritmos tienden a igualarse).

7.3 Líneas Futuras de Investigación

En el transcurso de la realización de esta tesis, han surgido algunas cuestiones que no hemos resuelto y que desarrollaremos en trabajos futuros. Destacamos las siguientes:

- La propuesta WP-TOPSISGA introduce problemas del mundo real que identifican una solución (Punto de Funcionamiento) estable y realizable de trabajo no necesariamente eficaz o Pareto óptima. Esta solución (definida en este trabajo

de tesis en el espacio de los objetivos) es utilizada para calcular los pesos usados en la ecuación de distancias por el método TOPSIS [101] y guiar la búsqueda hacia la región de interés (la región dominante del Punto de Funcionamiento). Pero hay que decir que si se conoce (en el espacio de los objetivos) el valor del Punto de Funcionamiento, también se conoce su valor de diseño en el espacio de las variables. En este caso, su inserción en la población inicial puede ser útil, pues como muestran los trabajos de Greiner et al. [90] y Winter et al. [211][212], permite ahorrar tiempo de cálculo y obtener soluciones mejoradas más fácilmente.

- Más arriba argumentamos algunas razones del porqué de nuestra elección de experimentar con problemas discretos. Sin embargo pensamos que podría ser útil extender los algoritmos que se proponen en esta tesis, a problemas en el dominio del continuo. En este sentido, primero decir que existe un trabajo de C.L. Hwang et al. [100] del año 1993, que extiende el método TOPSIS a problemas continuos y segundo, remarcar que los AEMOs trabajan con poblaciones y por tanto con conjuntos discretos de soluciones, por lo que los métodos aquí propuestos, podrían sin mucha dificultad ser adaptados para tratar problemas continuos.
- En su trabajo de tesis, D. Greiner [89] estudia métodos eficaces que permitan resolver la optimización multiobjetivo del diseño de pórticos metálicos (minimizando simultáneamente el peso restringido de la estructura y el número de distintos perfiles que conforman la misma). El autor relata las dificultades que existen con estos problemas, dado que el conjunto de soluciones (discretas) integrantes del frente óptimo de Pareto es de menor dimensión que el tamaño de la población no-dominada y la eficacia de los operadores eliminadores (por ejemplo el crowding distance -distancia de agrupamiento- de NSGAI [51][55] o el operador truncamiento de SPEAI [221]) se ve drásticamente limitada y propone el algoritmo DENSEA (Duplicate Elimination Non-dominated Sorting Evolutionary Algorithm). Resultados similares, se obtienen con los problemas discretos utilizados en esta tesis, cuando los tamaños de soluciones no-dominadas adoptados son próximos o superiores al tamaño real del frente óptimo del problema (los operadores similarity de TOPSISGA o WP-TOPSISGA, crowding distance de NSGAI, preference operator de R-NSGAI y similarity+crowding distance de CMOEA&TOPSIS o CMOEA20-80&TOPSIS, también ven fuertemente limitada su eficacia). Ponemos aquí de manifiesto, nuestro interés como trabajos futuros, del estudio de otros operadores eliminadores aplicados a problemas discretos con frentes de soluciones pequeños en comparación con el tamaño de la población no-dominada.

- En “A Reconciliation among Discrete Compromise Solutions”[213], K. Yoon mide la credibilidad de la función distancia d_p . Concluye que dicha función distancia es menos específica o menos creíble a medida que el valor de la métrica p se incrementa, por lo que recomienda utilizar el parámetro o métrica de valor $p=1$. Aunque, desde el punto de vista de la Teoría de la Utilidad Multiatributo [62][110] (en inglés MAUT), d_1 es la expresión de la forma aditiva simple de funciones de utilidad, la cual supone la existencia de una función de utilidad cardinal aditiva para los criterios que a su vez presupone: independencia entre los criterios y comparabilidad entre criterios de los valores de las alternativas (en otras palabras: lo que se pierde de un criterio se compensa con lo que se gana de otro), lo que no siempre se cumple y en cualquier caso es difícil de verificar. Todas las propuestas que hacemos en esta tesis, resuelven la problemática P_γ [161][163] y clasifican obteniendo un preorden completo las soluciones encontradas. Para ello, combinamos los AEMOs de segunda generación con el método TOPSIS. Sin embargo, TOPSIS necesita calcular las distancias d_p (lo que implica tener que elegir un valor de la métrica p) de cada solución con respecto a las soluciones Ideal y Anti-Ideal para así obtener el valor similaridad [45] en base al cual se clasifican las soluciones. En todas nuestras propuestas hemos utilizado la métrica $p=2$, aunque como describimos más arriba, al comparar los puntos de vista de K. Yoon y de la Teoría de la Utilidad Multiatributo, la elección del valor de dicha métrica p resulta comprometido. Los AEMOS a priori que proponemos, no resultan difíciles modificarlos e incorporar otros métodos existentes de Ayuda Multicriterio a la Decisión. En este sentido, proponemos utilizar el método muy conocido ELECTRE II [164], pues primero, elimina el problema antes mencionado de elección del valor de la métrica al utilizar el método TOPSIS, ya que se trata de un método de superación que reposa en el establecimiento de una relación de orden entre las diferentes soluciones, segundo, resuelve la problemática P_γ de clasificación de soluciones y tercero, no conocemos ningún trabajo sobre métodos evolutivos multiobjetivo que lo haya incorporado.

Bibliografía

- [1] S.F. Adra, I. Griffin, and P.J. Fleming. A Comparative Study of Progressive Preference Articulation Techniques for Multiobjective Optimisation. In S. Obayashi et al., editor, *EMO 2007, LNCS 4403*, pages 908–921. Springer-Verlag, Berlin Heidelberg, 2007.
- [2] S.E. Alonso. *Propuesta de un Método Evolutivo Flexible de Optimización Global*. PhD tesis, Universidad de Las Palmas de Gran Canaria, 2006.
- [3] J.D. Andrews and R.L. Patisson. Safety-system performance. pages 76–83. Philadelphia, Pennsylvania, 1997.
- [4] K.J. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, New York, 1951.
- [5] T. Bäck. Self-adaptation in Genetic Algorithms. In Francisco J. Varela y Paul Bourguine, editor, *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 263–271. Cambridge, MA. MIT Press., 1992.
- [6] T. Bäck. Optimal Mutation Rates in Genetic Search. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 2–8. Morgan Kaufmann Publishers, San Mateo, California, 1993.
- [7] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996.
- [8] J.E. Baker. Reducing bias and inefficiency in the selection algorithm. In J.J. Grefenstette, editor, *Proceedings of the 2nd. International Conference on Genetic Algorithms*, pages 14–21. Laurence Erlbaum Associates, 1987.
- [9] S. Barba-Romero and J.Ch. Pomerol. *Decisiones Multicriterio. Fundamentos Teóricos y utilización Práctica*. Servicio de Publicaciones de la Universidad de Alcalá, España, Madrid, 1997.
- [10] N.A. Barricelli. Esempi numerici di processi di evoluzione. *Methodos*, 1954.

- [11] M. Basseur and E-G. Talbi et al. Metaheuristics for multiobjective combinatorial optimization problems: Review and recent issues. Technical Report 5978, Institut National de Recherche en Informatique et en Automatique (INRIA), France, 2006.
- [12] R. Benayoun, B. Roy, and B. Sussman. Electre: une méthode pour guider le choix en présence des points de vue multiples. Technical Report 49, Direction Scientifique de la SEMA-METRA International, 1966.
- [13] J.C. Borda. Mémoire sur les élections au scrutin, Comptes rendus de l'Académie des sciences. Traduction: DE GRAZIA A., Mathematical derivation of a election system. *Isis*, 44:42–51, 1781.
- [14] D. Bouyssou and Ph. Vincke. *Concepts et méthodes pour l'aide à la décision 1: outils de modélisation. Sous la direction de D. Bouyssou, D. Dubois, M. Pirlot et H. Prade*. Lavoisier, Paris, 2006.
- [15] J. Branke and K. Deb. Integrating User Preferences into Evolutionary Multi-Objective Optimization. In Y. Jin, editor, *Knowledge Incorporation in Evolutionary Computation*, pages 461–477. Heidelberg, Germany: Springer, 2004.
- [16] J. Branke, T. KauBler, and H. Schmeck. Guidance in evolutionary multi-objective optimization. *Adv. Eng. Software*, 32(6):499–507, 2001.
- [17] J.P. Brans. *L'ingénierie de la décision. Elaboration d'instruments d'aide à la décision. Méthode PROMETHEE*, pages 182–213. Colloque d'aide à la décision. Université Laval, Québec, 1982.
- [18] J.P. Brans and B. Mareschal. How to select and how to rank projects: the PROMETHEE method for MCDM. *European Journal of Operations Research*, 24:228–238, 1986.
- [19] J.P. Brans and B. Mareschal. *Prométhée-GAIA: Une méthodologie d'aide à la décision en présence de critères multiples*. Éditions de l' Université de Bruxelles, Bruxelles, 2002.
- [20] J.P. Brans, B. Mareschal, and Ph. Vincke. Prométhée: a new family of outranking methods in multicriteria analysis. In J.P. Brans, editor, *MCDM, Operational Research '84*, pages 477–490. North-Holland, 1984.
- [21] J.P. Brans and Ph. Vincke. A preference ranking organization method. *Management Science*, 31:647–656, 1985.
- [22] H.J. Bremermann. The evolution of intelligence. the nervous system as a model of its environment. Technical Report 1, Contract No. 477(17), Department of Mathematics, University of Washington, Seattle, 1958.

- [23] A. Brindle. *Genetic Algorithms for Function Optimization*. PhD thesis, Department of Computer Science, University of Alberta, Edmonton, Alberta,, 1981.
- [24] R.L. Carraway, T.L. Morin, and H. Moskowitz. Generalized dynamic programming for multicriteria optimization. *European Journal of Operational Research*, 44:95–104, 1990.
- [25] A. Charnes and W.W. Cooper. *Management Models and Industrial Applications of Linear Programming*. John Wiley and Sons, New York, 1961.
- [26] A. Charnes, W.W. Cooper, and R. Ferguson. Optimal estimation of executive compensation by linear programming. *Management Science*, 1:138–151, 1955.
- [27] J.L. Cochrane and M. Zenely. *Multiple Criteria Decision Making*. University of South Carolina Press, Columbia, 1973.
- [28] R.F. Coelho, H. Bersini, and P. Bouillard. Parametrical mechanical design with constraints and preferences: application to a purge valve. *Comput. Methods Appl. Mech. Engrg.*, 192:4355–4378, 2003.
- [29] C.A. Coello. An updated survey of GA-based multiobjective optimization techniques. Technical Report Lania-RD-98-08, Laboratorio Nacional de Informatica Avanzada (LANIA), Xalapa, Veracruz, Mexico, 1998.
- [30] C.A. Coello. Using the Min-Max Method to Solve Multiobjective Optimization Problems with Genetic Algorithms. In *Lecture Notes in Computer Science*, volume 1484, pages 303–314. Springer-Verlag, London, UK, 1998.
- [31] C.A. Coello. A comprehensive survey of evolutionar-based multiobjective optimization techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, 1999.
- [32] C.A. Coello. Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. *IEEE Service Center*, 1:30–37, 2000.
- [33] C.A. Coello. *Introducción a la Computación Evolutiva (Notas de Curso)*. Departamento de Ingeniería Eléctrica -Sección Computación- (CINVESTAV-IPN), México, 2004.
- [34] C.A. Coello. Evolutionary Multi-Objective Optimization: A Historical View of the Field. *IEEE Computational Intelligence Magazine*, pages 28–36, 2006.
- [35] C.A. Coello and G.T. Pulido. Multiobjective optimization using a micro-genetic algorithm. In Lee Spector et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference, (GECCO'2001)*, pages 274–282. Morgan Kaufmann Publishers, San Francisco, California, 2001.

- [36] Y. Collette and P. Siarry. *Optimisation multiobjectif*. Éditions Eyrolles, Paris, 2002.
- [37] H.W. Corley. A new scalar equivalence for Pareto optimization. *IEEE Trans. Autom. Control*, 25:829–830, 1980.
- [38] D. Cvetković and C.A. Coello. Human Preferences and their Applications in Evolutionary Multi-Objective Optimisation. In Yaochu Jin, editor, *Studies in Fuzziness and Soft Computing: Knowledge Incorporation in Evolutionary Computation*, pages 479–502. Springer, 2005.
- [39] D. Cvetković and I. Parmee. Evolutionary design and multi-objective optimisation. pages 397–401. Aachen, Germany, 1998.
- [40] D. Cvetković and I. Parmee. Use of preferences for ga-based multi-objective optimisation. In W. Banzhafand et al., editor, *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1504–1510. Orlando, Florida, USA. Morgan Kaufmann, 1999.
- [41] D. Cvetković and I. Parmee. Designer’s Preferences and Multi-objective Preliminary Design Processes. pages 42–57. Berlin, Germany: Springer-Verlag, 2000.
- [42] D. Cvetković and I. Parmee. Preferences and their application in evolutionary multiobjective optimisation. *IEEE Transactions on Evolutionary Computation*, 6(1):42–57, 2002.
- [43] Ch.R. Darwin. *The Variation of Animals and Plants under Domestication*. Murray, London, second edition, UK, 1882.
- [44] Ch.R. Darwin. *The Origin of Species by Means of Natural Selection Or the Preservation of Favoured Races in the Struggle for Life*. Originally published in 1859. Cambridge University Press, Cambridge, UK, sixth edition, 1964.
- [45] B.V. Dasarathy. Smart: Similarity measured anchored ranking technique for the analysis of multidimensional data. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-6, 10:708–711, 1976.
- [46] J.A.N. Condorcet MARQUIS de CARITAT. *Essai sur l’application de l’analyse à la probabilité des décisions rendus à la pluralité des voix*. Imprimerie Royal, Paris, 1785.
- [47] M.A. De Vicente. *Ayuda Multicriterio a la decisión: problemática de los criterios en los métodos de sobreclasificación*. PhD tesis, Universidad Complutense de Madrid, 1998.

- [48] K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of the test problems. *Evolutionary Computation Journal*, 7(3):311–338, 1999.
- [49] K. Deb. Solving goal programming problems using multi-objective genetic algorithms. In *Proceedings of Congress on Evolutionary Computation*, pages 77–84, 1999.
- [50] K. Deb. Introducing bias among pareto-optimal solutions. In A. Ghosh and S. Tsutsui, editors, *Evolutionary Computing: Theory and Applications*, pages 263–292. London, Springer-Verlag, 2003.
- [51] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NS-GAII. Technical Report 200001, Indian Institute of Technology, Kanpur, 2000.
- [52] K. Deb, S. Chaudhuri, and K. Miettinen. Towards estimating nadir objective vector using evolutionary approaches. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2006)*, in press, 2006.
- [53] K. Deb and A. Kumar. Interactive Evolutionary Multi-Objective Optimization and Decision-Making using Reference Direction Method. Technical Report 2007001, Indian Institute of Technology, Kanpur, India, 2007.
- [54] K. Deb, M. Mohan, and S. Mishra. A Fast Multi-objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions. Technical Report 2003002, Indian Institute of Technology, Kanpur, India, 2003.
- [55] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. In *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [56] K. Deb and J. Sundar. Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2006)*, in press, 2006.
- [57] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. Technical Report 2001001, Indian Institute of Technology, Kanpur, 2001.
- [58] K.A. DeJong. *The Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, 1975.
- [59] J. Dréo, A. Pétrowski, P. Siarry, and É. Taillard. *Métaheuristiques pour l'optimisation difficile*. Éditions Eyrolles, Paris, 2003.

- [60] M. Ehrgott. A characterization of Lexicographic Max-ordering Solutions. In *Methods of Multicriteria Decision Theory: Proceedings of the 6th Workshop of the DGOR Working Group Multicriteria and Decision Theory*, pages 193–202, Egelsbach, Häsel-Hohenhausen, 1997.
- [61] M. Ehrgott. Multicriteria decision. In *Lecture Notes in Economics and Mathematical Systems*, volume 491. Springer, 2000.
- [62] P.C. Fishburn. *Utility theory for decision-making*. Wiley, New-York, 1970.
- [63] Cl. Flament. On incomplete preference structures. *Mathematical Social Sciences*, 5:61–72, 1983.
- [64] David B. Fogel. A analysis of evolutionary programming. In David B. Fogel and W. Atmar, editors, *Proceedings of the First Annual Conference on Evolutionary Programming*, pages 43–51. La Jolla, CA. Evolutionary Programming Society, 1992.
- [65] David B. Fogel. *Evolving artificial intelligence*. PhD tesis, University of California. San Diego, CA, 1992.
- [66] L.J.. Fogel. *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, 1966.
- [67] C.M. Fonseca, V. G. da Fonseca, and L. Paquete. Exploring the performance of stochastic multiobjective optimizers with the second-order attainment function. In C.A. Coello, A. Hernández, and E. Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005, Guanajuato, Mexico. volume 3410*, pages 221–234. Lecture Notes in Computer Science, Germany. Springer-Verlag, 2005.
- [68] C.M. Fonseca and P. J. Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. In H.M. Voigt et al., editor, *Parallel Problem Solving from Nature (PPSN-IV)*, pages 584–593. Berlin, Germany. Springer, 1996.
- [69] C.M. Fonseca and P.J. Fleming. Genetic Algorithms for Multi-objective Optimization: Formulation, Discussion and Generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423. San Mateo, California. Morgan Kauffman Publishers, 1993.
- [70] C.M. Fonseca and P.J. Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation. Spring*, 3(1):187–192, 1995.

- [71] C.M. Fonseca and P.J. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms i: A Unified Formulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(1):26–37, 1998.
- [72] C.M. Fonseca, J.D. Knowles, L. Thiele, and E. Zitzler. A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. In *An invited talk at the Evolutionary Multi-Criterion Optimization (EMO 2005), Guanajuato, Mexico*, 2005.
- [73] C.M. Fonseca, L. Paquete, and M. Lopez-Ibañez. An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. In *IEEE Congress on Evolutionary Computation*, pages 1157–1163. Vancouver, BC, Canada, 2006.
- [74] M.P. Fourman. Compaction of symbolic layout using genetic algorithms. In *Genetic Algorithms and their Applications Proceedings of the First Int. Conf. on Genetic Algorithms. Lawrence Erlbaum*, pages 141–153, 1985.
- [75] R.M. Friedberg. A Learning Machine: Part I. *IBM Journal of Research and Development*, 2(1):2–13, 1958.
- [76] R.M. Friedberg, B. Dunham, and J.H. North. A Learning Machine: Part II. *IBM Journal of Research and Development*, 3:282–287, 1959.
- [77] G.J. Friedman. *Selective Feedback Computers for Engineering Synthesis and Nervous System Analogy*. PhD thesis, University of California at Los Angeles, 1956.
- [78] B. Galván. *Contributions to Fault Tree Quantitative Evaluation*. PhD tesis, Physics Dep., Las Palmas de Gran Canaria University, Canary Islands, Spain, (In Spanish), 1999.
- [79] B. Galván, D. Salazar, D. Greiner, M. Méndez, and G. Winter. Optimización Conjunta del Diseño y la Estrategia de Mantenimiento de Sistemas de Seguridad en Industrias Potencialmente Peligrosas. *ESPOLCIENCIA, Ecuador*, Noviembre 2007.
- [80] B. Galván, G. Winter, D. Greiner, D. Salazar, and M. Méndez. New Evolutionary Methodologies for Integrated Safety System Design and Maintenance Optimization. In Springer Berlin/Heidelberg, editor, *Computational Intelligence in Reliability Engineering*, volume 39, pages 151–190, 2007.
- [81] X. Gandibleux, G. Libert, E. Cartignies, and P. Millot. SMART: Étude de la faisabilité d’un solveur de problèmes de mobilisation de réserve tertiaire. *Revue des systèmes de Décision*, 3(1):45–67, April 1994.

- [82] M. Gen and R. Cheng. *Genetic Algorithms and Engineering Design*. Wiley Interscience, John Wiley and Sons, USA, 1997.
- [83] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operational Research*, 5:533–549, 1986.
- [84] F. Glover and M. Laguna. *Tabu search*. Kluwer Academic Publishers, 1997.
- [85] D.E. Goldberg. *Genetic algorithms for search, optimization, and machine learning*. Reading, MA: Addison-Wesley, 1989.
- [86] D.E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In J.J. Grefenstette, editor, *Proceedings of the 2nd. International Conference on Genetic Algorithms*, pages 41–49. Lawrence Erlbaum Associates, 1987.
- [87] L. González, D. García, and B. Gálvan. An Intrinsic Order Criterion to Evaluate Large, Complex Fault Trees. *IEEE Transactions on Reliability*, 53(3):297–305, 2004.
- [88] John J. Grefenstette. Proceedings of the 1st. International Conference on Genetic Algorithms, Pittsburgh, USA. Lawrence Erlbaum Associates, 1985.
- [89] D. Greiner. *Optimización multiobjetivo de pórticos metálicos mediante algoritmos evolutivos*. PhD tesis, Computer Science, Civil Engineering and Applied Mathematics Department, Las Palmas de Gran Canaria University, Canary Islands, Spain, (In Spanish), 2005.
- [90] D. Greiner, J.M. Emperador, and G. Winter. Single and multiobjective frame optimization by evolutionary algorithms and the auto-adaptive rebirth operator. *Computer Methods in Applied Mechanics and Engineering*, 193:3711–3743, 2004.
- [91] D. Greiner, B. Galván, and G. Winter. Safety Systems Optimum Design by Multicriteria Evolutionary Algorithms. In *Evolutionary Multi-Criterion Optimization, LNCS 2632*, pages 722–736. Springer-Verlag, 2003.
- [92] D. Greiner, G. Winter, and B. Galván. Optimización Multiobjetivo en Sistemas de Seguridad: Una Comparativa entre los Algoritmos NSGAI y SPEA2. In B. Galván, G. Winter, P. Cuesta, and R. Aguasca, editors, *Actas del IV Congreso de Fiabilidad y Confiabilidad*, pages 156–165. Spain, 2002.
- [93] V. Grunert, C.M. Fonseca, and A.O. Hall. Inferential performance assessment of stochastic optimizers and the attainment function. In E. Zitzler et al., editor, *Evolutionary Multi-Criterion Optimization (EMO2001)*, pages 213–225. Berlin, Germany. Springer, 2001.

- [94] P. Hajela and C.Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 4, 1992.
- [95] A. Hallefjord and K. Jornsten. An entropy target-point approach to multiobjective programming. *Int. J. Syst. Sci.*, 17:639–653, 1986.
- [96] M.P. Hansen and A. Jaszkievicz. Evaluating the quality of approximations of the nondominated set. Technical Report IMM-REP-1998-7, Inst. of Mathematical Modeling, Tech. Univ. of Denmark, 1998.
- [97] J.H. Holland. Outline for a logical theory of adaptive systems. *Journal of the Association for Computing Machinery*, 9:297–314, 1962.
- [98] J.H. Holland. *Adaptation in natural and artificial systems*. PhD thesis, University of Michigan Press, 1975.
- [99] J. Horn. Multicriterion Decision Making and Evolutionary Computation. In Thomas Bäck, D. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation, volume 1*, pages F1.9:1–F1.9:15. IOP Publishing Ltd. and Oxford University Press, 1997.
- [100] C.L. Hwang, Y.J. Lai, and T.Y. Liu. A new approach for multiple objective decision making. *Computers and Operations Research*, 20(8):889–899, 1993.
- [101] C.L. Hwang and K. Yoon. *Multiple Attribute Decision Making: Methods and applications*. Springer-Verlag, Berlin, 1981.
- [102] J.P. Ignizio. *Goal Programming and Extensions*. Lexington Books, Massachusetts, 1976.
- [103] J.P. Ignizio. Review of Goal Programming: A Tool for Multiobjective Analysis. *Journal of the Operational Research Society*, 27:1109–1119, 1978.
- [104] H. Ishibuchi and T. Murata. A multi-objective genetic algorithm and its applications to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(3):392–403, 1998.
- [105] H. Ishibuchi and K. Narukawa. Performance Evaluation of Simple Multiobjective Genetic Local Search Algorithms on Multiobjective 0/1 Knapsack Problems. In *Proc. 2004 Congress on Evolutionary Computation (CEC'2004), volume 1*, pages 441–448. IEEE Service Center, 2004.
- [106] E. Jacquet-Lagrèze and J. Siskos. Assessing a set of additive utility functions for multicriteria decision making, the UTA method. *European Journal of Operational Research*, 10:151–164, 1982.

- [107] A. Jaszkiewicz. On the Performance of Multiple Objective Genetic Local Search on the 0/1 Knapsack Problem. A Comparative Experiment. *IEEE Transactions on Evolutionary Computation*, 6(4):402–412, 2002.
- [108] B.R. Jones, W.A. Crossley, and A.S. Lyrintzis. Aerodynamic and aeroacoustic optimization of airfoils via a parallel genetic algorithm. *Proc. of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, number AIAI-98-4811*, pages 1–11, 1998.
- [109] G. Jones and R.D. Brown et al. Searching databases of two-dimensional and three-dimensional chemical structures using genetic algorithms. In S. Forrest, editor, *Fifth International Conference on Genetic Algorithms*, pages 597–602, San Mateo, California, 1993. Morgan Kaufmann Publishers.
- [110] R.L. Keeney and H. Raiffa. *Decisions with multiple objectives: preferences and value tradeoffs*. Wiley, New-York, 1976.
- [111] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [112] J.D. Knowles and D.W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
- [113] J.D. Knowles and D.W. Corne. On metrics for comparing nondominated sets. In *Congress on Evolutionary Computation (CEC2002)*, pages 711–716. Piscataway, NJ: IEEE Press, 2002.
- [114] J.D. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical Report TIK-Report N° 214, Computer Engineering and Networks Laboratory, Zurich, 2005.
- [115] J.B. Kollat and P.M. Reed. The Value of Online Adaptive Search: A Performance Comparison of NSGAI, ϵ -NSGAI and ϵ -MOEA. In C.A. Coello, A. Hernández, and E. Zitzler, editors, *EMO 2005, LNCS 3410, Guanajuato, Mexico*, pages 386–398. Springer-Verlag, Berlin Heidelberg, 2005.
- [116] J.B. Kollat and P.M. Reed. Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design. *Advances in Water Resources. Elsevier*, 29:792–807, 2006.
- [117] P. Korhonen and J. Laakso. A visual interactive method for solving the multiple criteria problem. *European Journal of Operational Research*, 24:277–287, 1986.
- [118] J.R. Koza. Hierarchical genetic algorithms operating on populations of computer programs. In N. S. Sridharan, editor, *Proceedings of the 11th International*

- Joint Conference on Artificial Intelligence*, pages 768–774. Morgan Kaufmann, San Mateo, California, 1989.
- [119] J.R. Koza. *Programming. On the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge, Massachusetts, 1992.
- [120] D.H. Krantz, R.D. Luce, P. Suppes, and A. Tversky. *Foundations of measurement, vol. 1: Additive and polynomial representations*. Press, New-York, 1971.
- [121] H.W. Kuhn and A.W. Tucker. Nonlinear Programming. In *Proceedings of the second Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, Berkeley, 1951.
- [122] F. Kursawe. A variant of evolution strategies for vector optimization. In H.P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature. 1st Workshop PPSN I. Volume 496*, pages 193–197, Berlin. Germany, 1991. Lecture Notes in Computer Science., Springer-Verlag.
- [123] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, 10(3):263–282, 2002.
- [124] S.M. Lee. *Goal Programming For Decision Analysis*. Auerbach Publishers, Philadelphia, 1972.
- [125] D.H. Loughlin. *Genetic algorithm-based optimization in the development of tropospheric ozone control strategies*. PhD tesis, North Carolina State University, 1998.
- [126] D.H. Loughlin and S. Ranjithan. The neighborhood constraint method: A genetic algorithm-based multiobjective optimization technique. In T. Back, editor, *Seventh International Conference on Genetic Algorithms ICGA'97*, pages 666–677, San Mateo, California, 1997. Morgan Kaufmann Publishers.
- [127] R.D. Luce. Three axiom systems for additive semiordered structures. *SIAM Journal of Applied Mathematics*, 25:41–53, 1973.
- [128] F. Mallor-Giménez, R. Blanco, and C. Azcárate. Combining Linear Programming and Multiobjective Evolutionary Computation for Solving a Type of Stockastic Knapsack Problem. *Lecture Notes in Computer Science, EMO-07*, 4403:531–545, 2007.
- [129] J.A. Marglin. *Public Investment Criteria*. MIT Press, Cambridge, Massachusetts, 1967.
- [130] R.T. Marler and J.S. Arora. Survey of multi-objective optimization methods for engineering. *Springer-Verlag*, 26:369–395, 2004.

- [131] L.Y. Maystre, J. Pictet, and J. Simos. *Méthodes multicritère Electre. Description, conseils pratiques et cas d'application à la gestion environnementale*. Presses polytechniques et universitaires romandes, Lausanne, 1994.
- [132] G.J. Mendel. Versuche der Pflanzen-hybriden. *In Verhandlungen des naturforschenden Vereines in Brünn*, 1:3–47, 1865.
- [133] G.J. Mendel. Experiments in Plant Hybridisation (Traducción al inglés del artículo publicado originalmente en alemán en 1865). *Journal of the Royal Horticultural Society*, 26:1–32, 1901.
- [134] M. Méndez and B. Galván. Multiple-Objective Evolutionary Algorithms Using the Working Point and the TOPSIS Method. In R. Moreno et al., editor, *Lecture Notes in Computer Science (LNCS)*, volume 4739, pages 796–803. Springer-Verlag, Berlin Heidelberg, 2007.
- [135] M. Méndez, B. Galván, D. Salazar, and D. Greiner. Multiple-Objective Genetic Algorithm Using the Multiple Criteria Decision Making Method TOPSIS. (*To appear: Lecture Notes in Economics and Mathematical Systems (LNEMS), Springer*).
- [136] M. Méndez, B. Galván, and G. Winter. En las Cercanías del Punto de Funcionamiento Utilizando Algoritmos Genéticos Multiobjetivo. In *Actas del V Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2007)*, Tenerife, Febrero 2007.
- [137] M. Méndez, D. Salazar, and B. Galván. Optimización evolutiva multiobjetivo del diseño de Sistemas de Seguridad: Comparando el NSGAI y el criterio del tomador de decisiones. In *Actas del IV Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2005)*, Granada, Septiembre 2005.
- [138] A. Messac. Physical programming: effective optimization for computational design. *AIAA J.*, 34:149–158, 1996.
- [139] A. Messac and C.A. Mattson. Generating well-distributed sets of pareto points for engineering design using physical programming. *Optim. Eng.*, 3:431–450, 2002.
- [140] A. Messac, C. Puemi-Sukam, and E. Melachrinoudis. Mathematical and pragmatic perspectives of physical programming. *AIAA J.*, 39:885–893, 2001.
- [141] H. Meunier. *Algorithmes Évolutionnaires Parallèles pour l'Optimisation Multi-Objectif de Réseaux de Télécommunications Mobiles*. PhD thesis, Université des Sciences et Technologies de Lille, France, 2003.

- [142] K. Miettinen. *Nolinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.
- [143] P. Moscato. *On Evolution, Search, Optimazation, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*. Caltech Concurrent Computation Program, CP3 Report 826, 1989.
- [144] T. Murata and H. Ishibuchi. *MOGA: multi-objective algorithms*. In Proceedings of 1995 IEEE International Conference on Evolutionary Computation, Perth, WA, Australia, 1995.
- [145] J.F. Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.
- [146] J.F. Nash. Non cooperative games. *Annals of Mathematics*, 54:286–295, 1951.
- [147] Peter Nordin. A compiling genetic programming system that directly manipulates the machine code. In Kenneth E. Kinneer Jr., editor, *Advances in Genetic Programming*, pages 311–332. MIT Presss, 1994.
- [148] I. Othmani. *Optimisation multicritère : Fondements et Concepts*. PhD thesis, Université de Grenoble, France, 1998.
- [149] V. Pareto. *Cours D'Economie Politique, Volumen I et II, F. Rouge*. Lausanne, Suisse, 1896.
- [150] R.L. Pattisson and J.D. Andrews. Genetic algorithms in optimal safety system design. pages 187–197, 1999.
- [151] J. Periaux and M. Sefrioui. Nash Genetic Algorithms: Examples and Applications. In J. Periaux, P. Joly, O. Pironneau, and E. Oñate, editors, *Innovative Tools for Scientific Computation in Aeronautical Engineering*, pages 391–404. CIMNE, 2001.
- [152] A. Petrowski. A clearing procedure as a niching method for genetic algorithms. In T. Fukuda, editor, *Proccedings of the Third IEEE International Conference on Evolutionary Computation*, pages 798–803. IEEE press, 1996.
- [153] J.Ch. Pomerol and S. Barba-Romero. *Choix multicritère dans l' entreprise: principe et pratique*. Hermes, France, 1993.
- [154] I. Rechenberg. *Cybernetic Solution Path of an Experimental Problem*. Royal Aircraft Establishment Library Translation, 1965.
- [155] S. Ríos, C. Bielza, and A. Mateos. *Fundamentos de los Sistemas de Ayuda a la Decisión*. Ra-Ma Editorial, Madrid, 2006.

- [156] B.J. Ritzel, J.W. Eheart, and S. Ranjithan. Using genetic algorithms to solve a multiple objective groundwater pollution problem. *Water Resources Research*, 30(5):1589–1603, 1994.
- [157] M. Roubens. Preference relations on actions and criteria in multiple criteria decision making. *European Journal of Operational Research*, 10:51–55, 1981.
- [158] O. Roudenko. *Applications des Algorithmes Evolutionnaires aux Problèmes d’Optimisation Multi-Critère avec Contraintes*. PhD thesis, École Polytechnique, France, 2004.
- [159] B. Roy. Classement et choix en présence de points de vue multiples, la méthode ELECTRE. *R.I.R.O.*, 2(8):57–75, 1968.
- [160] B. Roy. Management scientifique et aide à la décision. Technical Report 86, Direction Scientifique de la SEMA, Paris, 1974.
- [161] B. Roy. Vers une méthodologie général d’aide à la décision. *Metra*, 14(3):459–497, 1975.
- [162] B. Roy. ELECTRE III: Algorithme de classement basé sur une représentation floue des préférences en présence de critères multiples. *Cahiers du C.E.R.O.*, 22(1):3–24, 1978.
- [163] B. Roy. *Méthodologie multicritère d’aide à la décision*. Economica, Paris, 1985.
- [164] B. Roy and P. Bertier. La méthode ELECTRE II. Une application au media-planning. In M. Ross, editor, *Operations Research72, Dublin 1972*, pages 291–302. North-Holland, 1973.
- [165] B. Roy and D. Bouyssou. *Aide à la décision: Méthodes et Cas*. Collection Gestion. Economica, Paris, 1993.
- [166] B. Roy and J. Hugonnard. Ranking of suburban line extension projects for the Paris metro system by a multicriteria method. *Transportation Research*, 16A:301–312, 1982.
- [167] B. Roy and J.M. Skalka. ELECTRE IS: aspects méthodologiques et guide d’utilisation. Document du LAMSADE 30, Université de Paris-Dauphine, Paris, 1984.
- [168] G. Rudolph. Convergence Analysis of Canonical Genetic Algorithms. *IEEE Transactions on Neural Networks*, 5:96–101, 1994.
- [169] T.L. Saaty. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15:234–281, 1977.
- [170] T.L. Saaty. *The Analytic Hierarchy Process*. McGrawHill, 1980.

- [171] D. Salazar, N. Carrasquero, and B. Gálvan. Exploiting Comparative Studies Using Criteria: Generating Knowledge from an Analyst's Perspective. In C.A. Coello, A. Hernández, and E. Zitzler, editors, *EMO 2005, LNCS 3410, Guanajuato, Mexico*, pages 221–234. Springer-Verlag, Berlin Heidelberg, 2005.
- [172] J. Sanchis, M.A. Martínez, and X. Blasco. Integrated multiobjective optimization and a priori preferences using genetic algorithms. *Information Sciences an International Journal, Elsevier Science*, 178(4):931–1262, 2008.
- [173] E. Sandgren. Advances in design optimization, Chapter: Multicriteria design optimization by goal programming. *Chapman and Hall, London*, pages 225–265, 1994.
- [174] J.D. Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD tesis, Vanderbilt University, 1984.
- [175] J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In Genetic Algorithms and their Applications, editors, *Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Lawrence Erlbaum, 1985.
- [176] J.R. Schott. *Fault tolerant desing using single and multi-criteria genetic algorithms*. Master's thesis, Department of Aeronautics and Astronautics, MIT, Boston. MA, 1995.
- [177] H-P. Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*. Birkhäuser, Basel, Alemania, 1977.
- [178] H-P. Schwefel. *Numerical Optimization of Computer Models*. Wiley, Chichester, UK, 1981.
- [179] M. Sefrioui and J. Periaux. Nash Genetic Algorithms: Examples and Applications. *Proceedings of the Congress on Evolutionary Computation, IEEE Press*, pages 509–516, 2000.
- [180] T. Sen, M.E. Raiszadeh, and P. Dileepan. A branch and bound approach to the bicriterion scheduling problem involving total flowtime and range of lateness. *Management Science*, 34(2):254–260, 1988.
- [181] A. Shärling. *Décider sur plusieurs critères, panorama de l'aide à la décision multicritère*. Presses polytechniques et universitaires romandes, Lausanne, 1985.
- [182] H.A. Simon. A behavioral model of rational choice. *Quarterly Journal of Economics*, 69:99–118, 1954.
- [183] H.A. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63:129–138, 1956.

- [184] V.P. Singh. *Applications des Algorithmes Évolutionnaires à la Détermination de Modèles de Vitesse par Inversion Sismique*. PhD tesis, Ecoles des Mines de Paris, 2006.
- [185] N. Srinivas and K. Deb. Multiobjective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation Journal*, pages 221–248, 1994.
- [186] B.S. Stewart and C.C. White. Multiobjective A*. *Journal of the ACM*, 38(4):775–814, 1991.
- [187] R. Storn and K. Price. Differential Evolution. A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.
- [188] P.D. Surry, N.J. Radcliffe, and I.D. Boyd. A Multi-Objective Approach to Constrained Optimisation of Gas Supply Networks: The COMOGA Method. In Terence C. Fogarty, editor, *Evolutionary Computing. AISB Workshop, Selected Papers*, pages 166–180, Sheffield, U.K., 1995. Lecture Notes in Computer Science, Springer-Verlag.
- [189] G. Syswerda. Uniform Crossover in Genetic Algorithms. In J.D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 2–9. Morgan Kaufmann Publishers, San Mateo, California, 1989.
- [190] G. Syswerda and J. Palmucci. The application of genetic algorithms to resource scheduling. In R.K. Belew and L.B. Booker, editors, *Fourth International Conference on Genetic Algorithms ICGA '4*, pages 502–508, San Mateo, California, 1991. Morgan Kaufmann Publishers.
- [191] K.C. Tan, E.F. Khor, T.H. Lee, and R. Sathikannan. An Evolutionary Algorithm with Advanced Goal and Priority Specification for Multi-objective Optimization. *Journal of Artificial Intelligence Research*, 18:183–215, 2003.
- [192] M. Tanaka and T. Tanino. Global optimization by the genetic algorithm in a multi-objective decision support system. *In Proceedings of the 10th. International Conference on Multiple criteria Decision making*, 2:261–270, 1992.
- [193] Y. Tang, P.M. Reed, and T. Wagener. How effective and efficient are multi-objective evolutionary algorithms at hydrologic model calibration? *Hydrology and Earth Systems Sciences*, 10:289–307, 2006.
- [194] A. Teller and M. Veloso. PADO: A new learning architecture for object recognition. In Katsushi Ikeuchi and Manuela Veloso, editors, *Symbolic Visual Learning*, pages 81–116. Oxford University Press, 1996.

- [195] Z. Tian and M.J. Zuo. Redundancy allocation for multi-state systems using physical programming and genetic algorithms. *Reliability Engineering & System Safety, Elsevier*, 91(9):1049–1056, 2006.
- [196] A. Tsoukiás. *De la théorie de la décision à l'aide à la décision*, pages 25–69. Concepts et méthodes pour l'aide à la décision, sous la direction de D. Bouyssou, D. Dubois, M. Pirlot et H. Prade. Lavoisier, Paris, 2006.
- [197] Ph. Vallin and D. Vanderpooten. *Aide à la Décision. Une approche par les cas*. Ellipses, France, 2002.
- [198] J.C. Vansnick. On the problem of weights in multiple criteria decision making (the noncompensatory approach). *European Journal of Operational Research*, 24:288–294, 1986.
- [199] D.A. Van Veldhuizen. *Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations*. PhD thesis, Graduate School of Eng. of the Air Force Inst. of Technol., Air Univ, 1999.
- [200] D.A. Van Veldhuizen and G.B. Lamont. Multiobjective Evolutionary Algorithm Research: A History and Analysis. Technical Report 98-03, Department of Electrical and Computer Engineering. Air Force Institute of Technology, USA, 1998.
- [201] D.A. Van Veldhuizen and G.B. Lamont. On measuring multiobjective evolutionary algorithm performance. In A. Zalzala and R. Eberhart, editors, *Congress on Evolutionary Computation (CEC2000), volume 1*, pages 204–211. Piscataway, NJ: IEEE Press, 2000.
- [202] D.A. Van Veldhuizen, B.S. Sandlin, and R.E. Marmelstein. Finding Improved Wire-Antena Geometries with Genetic Algorithms. *Fogel*, 28(3):102–107, 1997.
- [203] Ph. Vincke. *L'aide multicritère à la décision*. Éditions de l' Université de Bruxelles, Bruxelles, 1989.
- [204] M. Visée, J. Teghem, M. Pirlot, and E.L. Ulungu. Two-phases Method and Branch and Bound Procedures to Solve the Bi-objective Knapsack Problem. *Journal of Global Optimization*, 12:139–155, 1998.
- [205] A. Weismann. *The Germ Plasm: A Theory of Heredity*. Scott, London, UK, 1893.
- [206] A. Wetzel. *Evaluation of the effectiveness of genetic algorithms in combinatorial optimization*. University of Pittsburgh, Pittsburgh, 1983.
- [207] D. Whitley. The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 116–121, 1989.

- [208] P.B. Wienke, C. Lucasius, and G. Kateman. Multicriteria target optimization of analytical procedures using a genetic algorithm. *Analytical Chimica Acta* 265, (2):211–225, 1992.
- [209] A.P. Wierzbicki. The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Applications*, pages 468–486. Berlin, Germany. Springer-Verlag, 1980.
- [210] A.P. Wierzbicki. *Reference Point Approaches*. in: T. Gal, T.J. Stewart, T. Hanne (Eds.), *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications*, Kluwer Academic Publishers, Boston, 1999.
- [211] G. Winter, B. Galván, S. Alonso, and M. Méndez. New Trends in Evolutionary Optimization and its Impact on Dependability Problems. *EUROGEN, Munich*, Septiembre 2005.
- [212] G. Winter, D. Greiner, B. Galván, and B. González. Economical and environmental electric power dispatch optimization. In *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial and Sociatal Problems (EUROGEN), CIMNE 2003 Barcelona*, 2003.
- [213] K. Yoon. A Reconciliation among Discrete Compromise Solutions. In: *Journal of the Operational Research Society*, 38(3):277–286, 1987.
- [214] P.L. Yu and G. Leitmann. Compromise solutions, domination structures, and Salukvadze’s solution. *J. Optim. Theory Appl.*, 13:362–378, 1974.
- [215] W. Yu. *Aide multicritère à la décision dans le cadre de la problématique du tri: concepts, méthodes et applications*. PhD thesis, Université de Paris-Dauphine, France, 1992.
- [216] L.A. Zadeh. Optimality and Non-scalar Valued Performance Criteria. *IEEE Transactions on Automatic Control*, pages 59–60, 1963.
- [217] L.A. Zadeh. Fuzzy Sets. *Information and Control*, 8:338–353, 1965.
- [218] M. Zenely. *Multiple criteria decision making*. McGrawHill, New York, 1982.
- [219] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD tesis, Swiss Federal Institute of Technology Zurich, 1999.
- [220] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.

- [221] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In J. Periaux, P. Papailou, K. Giannakoglu, D. Tsahalis, and T. Fogarty, editors, *EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2002.
- [222] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms: A comparative case study. In A.E. Eiben et al., editor, *Parallel Problem Solving from Nature (PPSN-IV)*, pages 292–301. Berlin, Germany. Springer, 1998.
- [223] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.
- [224] E. Zitzler, L. Thiele, M. Laumanns, C.M Fonseca, and V. Grunert. Performance Assessment of Multiobjective Optimizers: A Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.