

UNIVERSIDAD DE GRANADA



Departamento de Ciencias de la Computación  
e Inteligencia Artificial

*Algoritmos Meméticos con  
Aplicación Adaptativa de la Búsqueda Local  
para Optimización Continua*

Tesis Doctoral

Daniel Molina Cabrera

Granada, Julio de 2007





UNIVERSIDAD DE GRANADA



**Algoritmos Meméticos con  
Aplicación Adaptativa de la Búsqueda Local  
para Optimización Continua**

MEMORIA QUE PRESENTA

Daniel Molina Cabrera

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

Julio de 2007

DIRECTORES

**Francisco Herrera Triguero y Manuel Lozano Márquez**

Departamento de Ciencias de la Computación  
e Inteligencia Artificial



La memoria titulada “*Algoritmos Meméticos con Aplicación Adaptativa de la Búsqueda Local para Optimización Continua*”, que presenta D. Daniel Molina Cabrera para optar al grado de doctor, ha sido realizada dentro del programa de doctorado “*Diseño, Análisis y Aplicaciones de Sistemas Inteligentes*” del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada, bajo la dirección de los doctores D. Francisco Herrera Triguero y D. Manuel Lozano Márquez.

Granada, Julio de 2007

El Doctorando

Los Directores

Fdo: Daniel Molina Cabrera

Fdo: F. Herrera Triguero y M. Lozano Márquez



Tesis Doctoral parcialmente subvencionada por la Comisión  
Interministerial de Ciencia y Tecnología con el proyecto  
TIC2005-08386-C05-01



**CICYT**

**TIC2005-08386-C05-01**



# Agradecimientos

Dedico esta memoria a mi madre, ya que le debo todo lo que soy, que nunca le podré agradecer lo bastante su apoyo y preocupación por mí.

Agradezco a mis directores y compañeros, Paco y Manolo, por haberme animado a empezar, y ayudar en una tarea, que aunque larga y dificultosa, se ha tornado finalmente posible y de la que sentirnos orgullosos. Les quiero en particular agradecer lo aprendido por trabajar a su lado, no tanto por los conocimientos sino por su actitud, tanto por su sentido crítico como por haberme transmitido su optimismo y ánimo en los momentos menos esperanzadores de este largo viaje.

Quiero agradecer a mis compañeros de fatigas que con sus consejos (o simplemente con su apoyo y confianza en los momentos que necesitaba desahogarme) me han hecho sentido acompañado: Carlos, José, Ramón, Rafa, Mariqui, Oscar, Jesús, y a todos los demás.

Quiero agradecerérselo también a los amigos que han evitado que la cabeza se me haya ido completamente, y que siempre han confiado en mí, que, a pesar de no llegar a entender lo que hacía, sí han entendido lo que significaba para mí y han estado conmigo: A mis amigos de siempre por haber seguido conmigo a pesar de prestarles cada vez menos tiempo; a mis amigos de Biblioteconomía por convertir de nuevo Granada en una ciudad acogedora y de descubrimientos para mí; y a mis amigos gaditanos y *quiritanas* por haberme hecho sentir más que acogido, querido, y por todo lo que he aprendido y descubierto a su lado; y por todos los buenos momentos que hemos pasado juntos.



# Indice

<b>Introducción</b>	<b>1</b>
A    Planteamiento . . . . .	1
B    Objetivos . . . . .	3
C    Resumen . . . . .	3
<b>1. Algoritmos Meméticos para Optimización Continua</b>	<b>5</b>
1.1. Optimización Continua . . . . .	6
1.2. Algoritmos Aproximativos . . . . .	7
1.2.1. Métodos de Búsqueda Local . . . . .	8
1.2.2. Algoritmos Evolutivos . . . . .	9
1.3. Algoritmos Meméticos . . . . .	12
1.3.1. Introducción a los Algoritmos Meméticos . . . . .	13
1.3.2. Enfoques de los Algoritmos Meméticos . . . . .	16
1.4. Búsqueda Global aportada por los Algoritmos Evolutivos . . . . .	18
1.5. Búsqueda Local y Explotación . . . . .	19
1.6. Mecanismo de Hibridación entre los Algoritmos Evolutivos y la Búsqueda Local . . . . .	21
1.7. Repaso Histórico de los Algoritmos Meméticos . . . . .	23
1.8. Ventajas de los Algoritmos Meméticos . . . . .	27

<b>2. Análisis de Algoritmos Evolutivos Híbridos</b>	<b>29</b>
2.1. Fomento de diversidad . . . . .	30
2.2. Modelos de Algoritmos Evolutivos para AMs . . . . .	32
2.2.1. El algoritmo CHC . . . . .	34
2.2.2. Búsqueda Dispersa ( <i>BD</i> ) . . . . .	37
2.2.3. Sistemas de Partículas: El modelo <i>PSO</i> . . . . .	40
2.2.4. Algoritmos Genéticos: Modelo Generacional y Estacionario	43
2.2.5. Operadores Comunes . . . . .	45
2.2.5.1. Operadores de cruce . . . . .	45
2.2.5.2. Operadores de mutación . . . . .	48
2.2.5.3. Métodos de Búsqueda Local Empleados . . . . .	50
2.2.5.4. Métodos de Selección de Padres . . . . .	53
2.2.5.5. Estrategias de Reemplazo . . . . .	56
2.3. Estudio Experimental . . . . .	58
2.3.1. Marco Experimental . . . . .	59
2.3.1.1. Características comunes de los distintos algoritmos	65
2.3.1.2. Tests Estadísticos . . . . .	65
2.3.1.3. Otras Medidas . . . . .	74
2.3.2. Estudio Individual de cada AE considerado . . . . .	74
2.3.2.1. Estudio Experimental del algoritmo CHC . . . . .	77
2.3.2.2. Estudio Experimental de la Búsqueda Dispersa . . . . .	84
2.3.2.3. Estudio Experimental del CLPSO . . . . .	90
2.3.2.4. Estudio Experimental del Algoritmo Genético Generacional . . . . .	91
2.3.2.5. Estudio Experimental del Algoritmo Genético Es- tacionario . . . . .	104
2.3.3. Análisis Comparativo entre los distintos AEs . . . . .	118
2.3.4. Conclusiones de la Experimentación . . . . .	126
2.4. Conclusiones . . . . .	128

<b>3. Algoritmo Memético con Búsqueda Local Basada en el Operador de Cruce (AMCR-<math>BLC^2P</math>)</b>	<b>131</b>
3.1. Operador de Cruce $BLX - \alpha$ Centrado en los Padres: $PBC - \alpha$	133
3.2. Método de Búsqueda Local Continuo Basado en el operador $PBC - \alpha$ : $BLC^2P$	134
3.3. Propuesta: AMCR- $BLC^2P$	136
3.3.1. Esquema AMCR general	137
3.3.2. Mecanismo Adaptable para Determinar $p_{BL}$	139
3.4. Marco Experimental	141
3.5. Análisis de los Componentes	142
3.5.1. Influencia de la Intensidad	142
3.5.2. Estudio de Sinergia entre $BLC^2P$ y $NAM$	144
3.5.3. Análisis de la Autoadaptación de $BLC^2P$	146
3.5.4. Estudio del Mecanismo Adaptable para $p_{BL}$	148
3.6. Estudio Comparativo Frente a otros AEs	150
3.6.1. Comparativa del AMCR- $BLC^2P$ frente a G3 y AG-FC	150
3.6.2. Comparativa del AMCR- $BLC^2P$ frente a otros modelos AMCRs	153
3.7. Conclusiones	156
<b>4. AMCR con Intensidad de BL Adaptativa</b>	<b>157</b>
4.1. Algoritmo Memético con Intensidad de Búsqueda Local Adaptativa	158
4.1.1. Componente de Búsqueda Local: CMA-ES	159
4.1.2. Memoria de la Búsqueda Local	161
4.1.3. Modelo de Hibridación	162
4.1.4. Ventajas del Modelo de Hibridación	164
4.2. Funciones de Evaluación y el Marco Experimental	164
4.2.1. Funciones de Test	165
4.2.2. Algoritmos del CEC'2005 comparados	166

4.3. Análisis de los Componentes de $MA^2LSI$ . . . . .	168
4.3.1. Categorías . . . . .	169
4.3.2. Ajustando los Parámetros de la BL: Ratio de Esfuerzo e Intensidad . . . . .	170
4.3.3. Influencia de la Memoria de Búsqueda Local . . . . .	174
4.3.4. Análisis del Comportamiento de CMA-ES como Búsqueda Local . . . . .	175
4.3.5. Estudio de la Intensidad de la Búsqueda Local, $n_{int}$ . . . . .	177
4.4. Análisis comparativo con las propuestas de la competición CEC 2005 . . . . .	179
4.4.1. $MA^2LSI$ Versus propuestas del CEC'2005, dimensión 10 . . . . .	180
4.4.2. $MA^2LSI$ Versus propuestas del CEC'2005, dimensión 30 . . . . .	184
4.4.3. $MA^2LSI$ Versus propuestas del CEC'2005, dimensión 50 . . . . .	189
4.5. Conclusiones . . . . .	192
<b>Comentarios Finales</b>	<b>193</b>
A. Resumen y Conclusiones . . . . .	193
B. Publicaciones Asociadas a la Tesis . . . . .	197
B. Líneas de Investigación Futuras . . . . .	198
<b>Apéndice</b>	<b>203</b>
<b>Bibliografía</b>	<b>244</b>

# Índice de figuras

1.1. Esquema de la Búsqueda Local . . . . .	8
1.2. Esquema de un AE . . . . .	10
1.3. Esquema de un AM . . . . .	15
2.1. Esquema de CHC . . . . .	36
2.2. Esquema de la Búsqueda Dispersa . . . . .	38
2.3. Seudocódigo de la Búsqueda Dispersa . . . . .	39
2.4. Estructura de un AGG . . . . .	43
2.5. Estructura del ciclo evolutivo de AGE . . . . .	44
2.6. Método de combinación $BLX - \alpha$ . . . . .	46
2.7. Método de combinación Aritmético . . . . .	46
2.8. Esquema del método de BL Solis Wets . . . . .	52
2.9. Operadores de Transformación para el método <i>Simplex</i> . . . . .	54
2.10. Esquema del Ranking Lineal . . . . .	55
2.11. Esquema del DC . . . . .	57
2.12. Esquema de la estrategia de reemplazo WAMS . . . . .	58
2.13. Ejemplo del test de Bonferroni-Dunn . . . . .	69
2.14. Ejemplo de Errores Acumulados . . . . .	75
2.15. Comparación CHC Sin BL y Solis Wets . . . . .	78
2.16. Comparación CHC Sin BL y Simplex . . . . .	80

2.17. Orden Medio para CHC con Solis Wets y Simplex . . . . .	83
2.18. Comparación BD Sin BL y Solis Wets . . . . .	86
2.19. Comparación BD Sin BL y Simplex . . . . .	87
2.20. Test de Bonferroni-Dunn con Búsqueda Dispersa (SW y S) . . . . .	89
2.21. Comparación de las Mejores variantes del AGG con Solis Wets . . . . .	97
2.22. Comparación de las mejores variantes del AGG con Simplex . . . . .	101
2.23. Test de Bonferroni-Dunn sobre distintos AGGs (SW y S) . . . . .	103
2.24. Test de Bonferroni-Dunn sobre el AGE Sin BL . . . . .	107
2.25. Test de Bonferroni-Dunn sobre el AGE SW 10 . . . . .	107
2.26. Test de Bonferroni-Dunn sobre el AGE SW 50 . . . . .	108
2.27. Test de Bonferroni-Dunn sobre el AGE SW 100 . . . . .	108
2.28. Comparación variantes del AGE con BLX-0.5, sin BL . . . . .	110
2.29. Comparación variantes del AGE con BLX-0.5, con Solis Wets, $n_{int}=10$ . . . . .	112
2.30. Comparación variantes del AGE con BLX-0.5, con Solis Wets, $n_{int} = 50$ . . . . .	113
2.31. Comparación variantes del AGE con BLX-0.5, con Solis Wets, $n_{int}=100$ . . . . .	115
2.32. Orden Medio de las mejores AGEs con BLX-0.5 y SW, calculado para cada valor de intensidad . . . . .	117
2.33. Comparación de las mejores variantes de los AEs sin BL . . . . .	119
2.34. Comparación de las mejores variantes de los AEs, con Solis Wets y $n_{int}=10$ . . . . .	121
2.35. Comparación de las mejores variantes de los AEs, con Solis Wets y $n_{int}=50$ . . . . .	123
2.36. Comparación de las mejores variantes de los AEs, con Solis Wets y $n_{int}=100$ . . . . .	125
2.37. Orden Medio de los mejores AGEs, calculados por intensidad . . . . .	127
3.1. Operador de cruce $PBC - \alpha$ . . . . .	134

---

3.2.	Esquema de $BLC^2P$ . . . . .	135
3.3.	AMCR estacionario propuesto . . . . .	137
3.4.	Ratio $\frac{L}{G}$ producida por el AMCR- $BLC^2P$ . . . . .	149
4.1.	Comparación de las distintas combinaciones $MA^2LSI(ratio_{BL}, n_{int})$ en multimodales, dimensión 10 . . . . .	171
4.2.	Comparación de las distintas combinaciones $MA^2LSI(ratio_{BL}, n_{int})$ en multimodales, dimensión 30 . . . . .	172
4.3.	Comparación de las distintas combinaciones $MA^2LSI(ratio_{BL}, n_{int})$ en multimodales, dimensión 50 . . . . .	173
4.4.	Restricción del espacio de búsqueda . . . . .	176
4.5.	Porcentaje de veces que se aplica la BL a una nueva solución (no mejorada anteriormente) . . . . .	178
4.6.	Número de veces que se aplica la BL a soluciones mejoradas . . . . .	179
4.7.	Test de Bonferroni-Dunn sobre CEC'2005 y $MA^2LSI(0.8, 100)$ , dimensión 10 . . . . .	181
4.8.	Error Acumulado Medio sobre CEC'2005 y $MA^2LSI(0.8, 100)$ , dimensión 10 . . . . .	182
4.9.	Test de Bonferroni-Dunn sobre CEC'2005 y $MA^2LSI(0.5, 500)$ , con dimensión 30 . . . . .	186
4.10.	Error Acumulado Medio sobre CEC'2005 y $MA^2LSI(0.5, 500)$ , dimensión 30 . . . . .	187



# Indice de tablas

2.1. Definición de las Funciones de Tests . . . . .	61
2.2. Dominios de las Funciones de Tests . . . . .	62
2.3. Matrices del Sistema de Ecuaciones Lineales . . . . .	62
2.4. Ejemplo de test Iman-Davenport . . . . .	68
2.5. Test de Holm sobre el algoritmo A1 . . . . .	71
2.6. Ejemplo del test de Wilcoxon Test usando A1 como referencia . . .	73
2.7. Test de Iman-Davenport con CHC . . . . .	77
2.8. Test de Iman-Davenport con CHC (SW y S) . . . . .	81
2.9. Test de Iman-Davenport con Búsqueda Dispersa . . . . .	84
2.10. Test de Iman-Davenport con BD (SW y S) . . . . .	88
2.11. Test de Iman-Davenport con CLPSO . . . . .	91
2.12. Diferencia de EAc entre las variantes con la mutación No Uniforme y la BGA para AGG con SW . . . . .	93
2.13. Test de Iman-Davenport con AGG y BGA . . . . .	94
2.14. Test de Holm de AGG Versus mejor(TS-BLX-0.5) con Solis Wets, sin BL y SW 10 . . . . .	95
2.15. Test de Holm de AGG Versus mejor(TS-BLX-0.5) con Solis Wets, con SW 50 y 100 . . . . .	96
2.16. Test de Holm de AGG Versus mejor(TS-BLX-0.5) con Simplex, sin BL y con S 10 . . . . .	99

2.17. Test de Holm de AGG Versus mejor(TS-BLX-0.5) con Simplex, con S 50 y S 100 . . . . .	100
2.18. Test de Iman-Davenport con AGG (SW y S) . . . . .	102
2.19. Errores Acumulados Mutación No Uniforme Versus BGA, para AGE con SW . . . . .	105
2.20. Test de Iman-Davenport para AGE y Solis Wets, agrupados por su estrategia de reemplazo . . . . .	106
2.21. Test de Iman-Davenport para AGE con BLX-0.5 . . . . .	109
2.22. Mejores Combinaciones de Selección-Reemplazo con el AGE . . . . .	116
2.23. Test de Iman-Davenport de las mejores propuestas . . . . .	118
2.24. Test de Holm de los distintos AEs respecto al mejor: AGE (TS-DC-BLX-0.5), sin BL . . . . .	120
2.25. Test de Holm de los distintos AEs respecto al mejor: AGE (NAM-RW-BLX-0.5) (SW 10) . . . . .	122
2.26. Test de Holm de los distintos AEs respecto al mejor: AGE (NAM-RW-BLX-0.5) (SW 50) . . . . .	124
2.27. Test de Holm de los distintos AEs respecto al mejor: CHC (BLX-0.5) (SW 100) . . . . .	126
2.28. Resumen de las Mejores Variantes para cada valor de profundidad . . . . .	127
3.1. Resultados de AMCR- $BLC^2P$ con diferentes valores para $n_{it}$ . . . . .	144
3.2. Resultados con diferentes combinaciones de $NAM$ y $BLC^2P$ . . . . .	146
3.3. Comparación entre AMCR- $BLC^2P$ y AMCR-MD . . . . .	147
3.4. Comparación con diferentes valores fijos para $p_{BL}$ Versus esquema adaptativo del AMCR- $BLC^2P$ . . . . .	150
3.5. Comparación de AMCR- $BLC^2P$ con los algoritmos G3 y el AG-FC, funciones 1-6 (I) . . . . .	152
3.6. Comparación de AMCR- $BLC^2P$ con los algoritmos G3 y el AG-FC, funciones 7-8 (II) . . . . .	153
3.7. Comparación de AMCR- $BLC^2P$ versus otros AMCRs . . . . .	155

4.1. Test de Iman-Davenport de $MA^L SI(ratio_{BL}, n_{int})$ para cada dimensión, funciones multimodales . . . . .	170
4.2. Test de Wilcoxon Sin Memoria Versus Con Memoria . . . . .	175
4.3. Test de Wilcoxon <i>Con Restricciones</i> Versus <i>Sin Restricciones</i> , funciones multimodales . . . . .	177
4.4. Test de Iman-Davenport con CEC2005 y MA2LSI(0.8,100) para dimensión 10 . . . . .	180
4.5. Test de Wilcoxon CEC'2005 Versus MA2LSI(0.8,100), dimensión 10 . . . . .	183
4.6. Resumen del test de Wilcoxon CEC'2005 VS MA2LSI(0.8,100), con dimensión 10 . . . . .	184
4.7. Test de Iman-Davenport con CEC2005 y MA2LSI(0.5,500) para dimensión 30 . . . . .	185
4.8. test de Wilcoxon CEC'2005 VS MA2LSI(0.5,500), dimensión 30 . . . . .	188
4.9. Resumen del test de Wilcoxon CEC'2005 VS MA2LSI(0.5,500), dimensión 30 . . . . .	188
4.10. Test de Wilcoxon CEC'2005 VS MA2LSI(0.5,1000), dimensión 50 . . . . .	190
4.11. Test de Wilcoxon L-CMA-ES Versus MA2LSI . . . . .	190
4.12. Test de Wilcoxon G-CMA-ES Versus MA2LSI . . . . .	191
4.13. Valores de Error Alcanzados Con FES=1e5 para Problemas 1-16, dimensión 10 . . . . .	204
4.14. Valores de Error Alcanzados Con FES=1e5 para Problemas 17-25, dimensión 10 . . . . .	205
4.15. Valores de Error Alcanzados Con FES=3e5 para Problemas 1-16, dimensión 30 . . . . .	206
4.16. Valores de Error Alcanzados Con FES=3e5 para Problemas 17-25, dimensión 30 . . . . .	207
4.17. Valores de Error Alcanzados Con FES=5e5 para Problemas 1-16, dimensión 50 . . . . .	208
4.18. Valores de Error Alcanzados Con FES=5e5 para Problemas 17-25, dimensión 50 . . . . .	209



# Introducción

## A Planteamiento

La optimización es, actualmente, un área que recibe mucho interés debido a que se presenta en multitud de problemas de ciencia e ingeniería que, por su propia naturaleza, no pueden ser resueltos mediante un procedimiento de búsqueda exhaustivo (que obtenga la mejor solución tras una búsqueda por todo el dominio de posibles valores).

Los problemas de optimización complejos están siendo abordados mediante el uso de algoritmos aproximados. Dentro de los algoritmos aproximados hay dos enfoques muy distintos.

- El empleo de mecanismos de explotación de los resultados de la búsqueda, como los procedimientos de Búsqueda Local (BL), que permiten alcanzar rápidamente soluciones precisas. El principal inconveniente que presentan estos métodos es que poseen mucha tendencia a quedarse atrapados en óptimos locales y no en el óptimo global. Mucho trabajo se ha realizado en este sentido [Sch81, GK03].
- El empleo de mecanismos de exploración, como los algoritmos evolutivos (AE) [ES03], entre ellos los algoritmos genéticos (AGs) [Gol89]. Estos algoritmos permiten, dado un dominio de exploración, realizar una amplia búsqueda, permitiendo no quedar atrapados en óptimos locales. Pero, por otro lado, presentan problemas para alcanzar soluciones precisas rápidamente.

Frente a estos dos criterios, en los últimos años se han planteado algoritmos que buscan abordar el problema de la búsqueda (búsqueda global pero con capacidad para alcanzar soluciones precisas rápidamente) combinando ambos enfoques. Estos algoritmos, llamados Algoritmos Meméticos (AMs) [Mos99] combinan un método de evolución de poblaciones (como un AE, con características de búsqueda global) con un método de mejora de las soluciones encontradas, como un algoritmo de BL, para obtener una sinergia entre ambos componentes, consiguiendo las ventajas de ambos compensando al mismo tiempo las deficiencias individuales de cada uno.

Dentro de los problemas de optimización, existe una categoría, la de optimización en dominios continuos, en la que los AMs se presentan especialmente prometedores, ya que en éstos la necesidad de equilibrio entre soluciones precisas (mediante ajuste fino de las variables) y exploración global es aún mayor, si cabe, que en optimización en dominios discretos. Los AMs para este tipo de problemas los vamos a denominar Algoritmos Meméticos con Codificación Real (AMCR).

Un AM se compone, principalmente, de tres componentes:

- Un AE, encargado del componente explorativo de la búsqueda. Aunque usualmente se emplea un AE clásico, al emplearse conjuntamente con una BL no posee la misma necesidad de explotación que un AE que se utilice sin hacer uso de dicha BL.
- Un método de BL, encargado de explotar las soluciones obtenidas por el AE. La BL permite mejorar la precisión de las soluciones rápidamente, pero puede orientar rápidamente la búsqueda hacia óptimos locales, y no hacia el óptimo global.
- Un mecanismo de hibridación, que determina la relación entre el AE y la BL, estableciendo a qué individuos se le aplicará la BL, y las características de dicha aplicación.

Para poder obtener un buen resultado es necesario un equilibrio entre la exploración del AE y la explotación debida a la BL, para lo cual ambos algoritmos deben de complementarse. El uso del mecanismo de hibridación es crucial para un correcto desempeño. Dentro de los mecanismos de hibridación, son especialmente interesantes aquellos que utilizan la información de los propios individuos para elegir aquellos que recibirán la atención de la BL.

En nuestra memoria estudiaremos el uso de AMs para resolver problemas de optimización continua, para lo cual pondremos distintos mecanismos de hibridación que aplican la BL de forma adaptativa.

## B Objetivos

Como se acaba de mencionar, el principal objetivo de esta memoria es abordar el diseño de un adecuado AMCR para problemas de optimización continua, planteando el diseño de cada uno de los componentes del AM.

En concreto, el objetivo propuesto se subdivide en los siguientes objetivos concretos:

- Evaluar distintos esquemas de AEs como algoritmos de exploración candidatos para un AM. Analizamos su comportamiento tras incorporar un método de BL, ya que no nos interesa los AEs que presentan sólo un buen comportamiento de forma aislada, si no aquellos que presentan un buen comportamiento tras combinarse con un método de BL.
- Proponer diferentes tipos de BL para su uso en el AM.
- Proponer modelos de hibridación adaptativos que utilicen las características de una solución para determinar si se aplicará la BL sobre ésta y en qué grado, para orientar la BL hacia las soluciones más prometedoras.

## C Resumen

La memoria está dividida en cuatro capítulos que se describen brevemente a continuación.

En el Capítulo 1, presentaremos la problemática de la optimización continua y los AMs y cómo encaja el uso de éstos dentro de dicha problemática. Posteriormente, estudiaremos sus distintos componentes, presentando conceptos importantes para su buen rendimiento. Haremos un repaso histórico de distintos AMs propuestos en la literatura para optimización con parámetros reales, con

especial énfasis en los AMs adaptativos. Finalmente, haremos un resumen de sus principales ventajas.

En el Capítulo 2, planteamos que el uso de una BL dentro de un AE modifica su comportamiento, por lo que se hace necesario un estudio comparativo entre distintos AEs que considere la diferencia de su comportamiento conforme se aumenta la explotación mediante una BL. En este capítulo presentaremos un análisis de un conjunto, pequeño, pero significativo, de AEs, para determinar aquellos que se perfilen como buenos candidatos. Primero, se presentan los distintos algoritmos considerados, y luego, el estudio experimental desarrollado, donde se especifica la metodología seguida, sus resultados y un completo análisis de los mismos.

En el Capítulo 3, proponemos un AMCR combinando el AE recomendado en el capítulo anterior, y un modelo propuesto de BL basado en un operador de cruce centrado en el padre, capaz de aportar alta presión explotativa con pocas iteraciones. Para orientar la BL sobre las soluciones más prometedoras, proponemos un nuevo esquema de hibridación adaptativo que determina la aplicación o uso de la BL sobre una solución en función de su valor para la función objetivo. Posteriormente, se detallará el estudio experimental desarrollado, donde se especifica la metodología seguida, sus resultados y un completo análisis de los mismos.

Tras observar la conveniencia de adaptar la aplicación de la BL sobre un individuo en base a las características de dicho individuo, en el Capítulo 4 se propone un nuevo modelo de hibridación que aplica la BL con una intensidad adaptativa en función del propio individuo (mayor a los más prometedores). Este proceso implica un criterio de selección adecuado para elegir los individuos a mejorar, y también para decidir cuánta intensidad debe de aplicarse en cada caso. Este enfoque está orientado para sistemas de BL que requieren una alta intensidad. Este modelo de hibridación se usará para diseñar un nuevo AM combinando el AE seleccionado en el capítulo 2, usando el algoritmo CMA-ES como método de BL. Posteriormente, se detallará el estudio experimental desarrollado, donde se especifica la metodología seguida, sus resultados y un completo análisis de los mismos.

Incluimos una sección de *Comentarios Finales*, que resume los resultados obtenidos en esta memoria, presentando algunas conclusiones sobre éstos. Finalmente, se comentarán algunos aspectos sobre trabajos futuros que quedan abiertos en la presente memoria.

# Capítulo 1

## Algoritmos Meméticos para Optimización Continua

Muchos problemas de ingeniería pueden expresarse como problemas de optimización continua. En estos problemas, el encontrar un óptimo global, debido a su propia naturaleza intratable, no es una tarea que pueda resolverse numéricamente de forma exacta, por lo que se hace necesario abordar este tipo de problemas mediante técnicas aproximativas, que intentan obtener una solución lo más adecuada posible dentro de unos límites de evaluación y/o temporales.

En los problemas de optimización existen dos importantes factores de la búsqueda [YI01]:

- La exploración, también llamada diversificación en la literatura, es el proceso de guiar la búsqueda hacia regiones no exploradas. Un algoritmo que realizase una insuficiente exploración podría obviar regiones enteras, por lo que si el óptimo se encontrase en una de esas regiones no tendría ninguna posibilidad de ser encontrado.
- La explotación, también llamada intensificación en la literatura, es el proceso de realizar una búsqueda de mejores soluciones minucioso e intenso en un entorno cercano a buenas soluciones ya encontradas (considerando que son zonas más prometedoras). Este proceso es fundamental para obtener

una mayor precisión en las soluciones encontradas.

Los algoritmos meméticos (AMs) [Mos99] [MC03] son un tipo de procedimientos que combinan dos componentes, cada uno responsable principalmente de cada uno de estos factores. El objetivo de los AMs es el de hacer que ambos elementos trabajen de forma cooperativa para conseguir una continua sinergia entre ambos que permita mejorar el proceso de búsqueda [Har94].

En este capítulo introducimos el problema de optimización continua, y presentamos los algoritmos AMs como un tipo de algoritmo especialmente indicado para abordarlos, analizando los distintos elementos que lo componen y sus propiedades más interesantes. El objeto de este capítulo es presentar los distintos conceptos que serán de gran ayuda para el desarrollo del resto de capítulos de esta memoria.

Este capítulo presenta la siguiente estructura: En la Sección 1.1 presentamos el problema de optimización continua. En la Sección 1.2 presentamos los distintos enfoques aproximativos empleados para abordarlo: los métodos de BL, y los AEs. En la Sección 1.3 se presentan los AMs, definiéndonos y mostrando su enfoque. En las tres secciones sucesivas se describen los distintos elementos que lo componen y su influencia: En la Sección 1.4, el algoritmo encargado de la búsqueda global; en la Sección 1.5, el método de BL; y en la Sección 1.6, el mecanismo de hibridación empleado para combinar ambos elementos. En la Sección 1.7 se realiza un repaso histórico de los principales hitos en la existencia AMs. Finalmente, en la Sección 1.8 se detallan algunas de las principales ventajas del empleo de los AMs en los problemas de optimización continua.

## 1.1. Optimización Continua

Muchos problemas de ingeniería pueden formularse como problemas de optimización sobre una función objetivo definida sobre un dominio  $D$  de soluciones alcanzables. Resolverlos supone encontrar dentro de un dominio definido la solución que permita minimizar dicha función objetivo (óptimo global). Dicha problemática se presenta en numerosos problemas de investigación e ingeniería, cuando se necesita obtener la mejor de un conjunto de soluciones a un problema dado, y se posee un criterio capaz de medir la calidad de cada solución. En los problemas en los que se desea maximizar una función objetivo  $f$  pueden transformarse dicho objetivo en minimizar otra función  $g$ , sin pérdida de generalidad.

Matemáticamente, se puede expresar que, dada una función  $f : D \rightarrow \mathbb{R}$ , el objetivo del óptimo global es encontrar  $x^*$  para el que:

$$x^* = \operatorname{argmin}_{x \in D} f(x)$$

En esta memoria nos centraremos en la problemática de optimización continua (es decir, en la que  $D \subseteq \mathbb{R}^N$ ). Para dicho dominio, el problema de encontrar el óptimo global se puede redefinir en, dado un  $\epsilon \in \mathbb{R}, \epsilon > 0$ , encontrar un valor  $x'$  tal que

$$f(x^*) - f(x') \leq \epsilon$$

## 1.2. Algoritmos Aproximativos

Para cualquiera de las dos expresiones anteriores, el encontrar el óptimo global no es un problema resoluble numéricamente de forma exacta, debido a la imposibilidad de poder explorar la totalidad del conjunto de soluciones posibles. Por lo tanto, para abordar este tipo de problemas se utilizan técnicas aproximativas.

En estas técnicas no exhaustivas se representa cada solución mediante un individuo, que representa un punto de búsqueda dentro de un espacio de un conjunto de soluciones potenciales del problema. Cada uno de estos individuos posee asociado un valor de bondad, determinado por el valor que toma sobre dicho individuo una *función objetivo* (normalmente, la función que se desea optimizar).

El proceso que siguen estas técnicas es partir de una solución o de un conjunto de soluciones de partida, obtenidas aleatoriamente o mediante un método de generación de soluciones iniciales. Dicho conjunto de soluciones constituye lo que se denomina una población de individuos, que evolucionará hasta obtener un conjunto de soluciones más optimizadas que el inicial. Dicha evolución se realizará mediante un proceso iterativo que genera nuevas soluciones (mediante modificación de las anteriores o combinación de éstas), que serán evaluadas y, eventualmente, serán introducidos en la población. Cuando haya transcurrido un tiempo máximo prefijado, o evaluado un número máximo de soluciones, el proceso termina y devuelve la mejor o las mejores soluciones encontradas hasta ese momento. Al no evaluar todo el espacio de posibles soluciones no se puede garantizar que la mejor solución encontrada sea el óptimo global buscado.

Así pues, estas técnicas no exhaustivas permiten obtener la mejor solución dentro de unas restricciones, de tiempo o de evaluaciones, mediante la búsqueda de un compromiso entre la bondad de la solución encontrada y el esfuerzo invertido para ello.

A continuación, comentamos dos enfoques aproximativos muy distintos: Los métodos de búsqueda local (BL) y los algoritmos evolutivos (AEs).

### 1.2.1. Métodos de Búsqueda Local

Los métodos de BL [Sch81] son métodos que definen un proceso de transformación que, partiendo de una solución determinada, generan una o varias soluciones en un entorno cercano. Así pues, partiendo de una primera solución, denominada *solución actual*, se puede obtener un conjunto de soluciones directamente alcanzables mediante una única aplicación de dicho método. A este conjunto de soluciones accesibles se les denomina *vecindario* de la primera solución. Para obtener una solución adecuada, estas técnicas siguen un proceso iterativo en el que, dada una solución actual, buscan una mejor solución dentro de su vecindario que se convierte en la nueva solución actual. Este proceso continúa hasta que se genera un número máximo de soluciones, o se alcanza un óptimo local. Un óptimo local es una solución para la cual no existe ninguna solución dentro de su vecindario que la mejore. Es decir, es una solución  $x$  tal que  $f(x) = \min f(y) | \forall y \in \text{vecindario}(x)$ . La Figura 1.1 representa un esquema común de la BL en el que la nueva solución sustituye a la anterior sólo si la mejora.

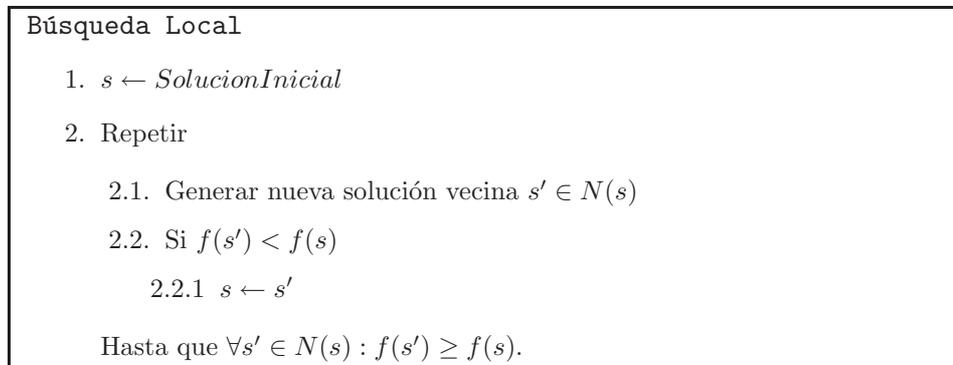


Figura 1.1: Esquema de la Búsqueda Local

Una ventaja de la BL sobre otras heurísticas es la eficiencia de la búsqueda. En muchas situaciones, se puede calcular utilizando información del problema de forma muy eficiente  $\nabla f = f(\nabla s)$  comparando las diferencias entre las soluciones  $s$  y  $s'$ , para posteriormente calcular  $f(s') = f(s) + \nabla f$ .

Un inconveniente de la BL es que, por propia definición, está orientado a alcanzar óptimos locales (las mejores solución dentro de una zona pequeña, el vecindario) en vez del objetivo deseado, que es alcanzar el óptimo global (la mejor solución dentro de todo el espacio de búsqueda). Además, por el propio procedimiento de la búsqueda local, estos métodos pueden converger de forma muy rápida a los óptimos locales, que pueden ser soluciones muy alejadas del óptimo global. A este comportamiento se denomina convergencia prematura [ES91].

Al esquema comentado anteriormente, de reemplazar la solución actual sólo si la nueva lo mejora, se le denomina *ascensión de colinas*, y es ampliamente utilizado. Existe, sin embargo, otros enfoques de búsqueda, como la búsqueda tabú [GL97] o el enfriamiento simulado [KGV83], que permiten que la nueva solución actual pueda ser peor que la anterior, permitiendo que en ocasiones pueda evitar quedarse estancado el proceso en un óptimo local. En nuestra memoria, sin embargo, nos centraremos en el primer esquema, ya que es más empleado en dominio continuo.

### 1.2.2. Algoritmos Evolutivos

Los algoritmos evolutivos (AEs) [BFM97] fueron inspirados por la teoría de la evolución de Darwin [Dar59]. Según su teoría, la evolución de las especies se produce por tres conceptos: *replicación*, *variación* y *selección natural*. La población se mantiene mediante replicación, y la variación introduce diversidad en la población, mediante una serie de diferencias entre los descendientes y su progenitores (generando nuevos individuos aplicando un mecanismo de recombinación sexual de dos individuos, en el que el nuevo individuo hereda características de ambos). Debido a los recursos limitados y a agentes externos (depredadores), este proceso se desarrolla en un ambiente hostil en el que los individuos de la misma o distintas especies compiten entre sí y sólo los más adaptados sobreviven. Esto implica que los más adaptados tendrán mayores posibilidades de tener descendencia, y por tanto, de dejar su herencia genética (sus características) a las generaciones consecutivas, conduciendo a un proceso en el que la población va evolucionando obteniendo cada vez mejores individuos.

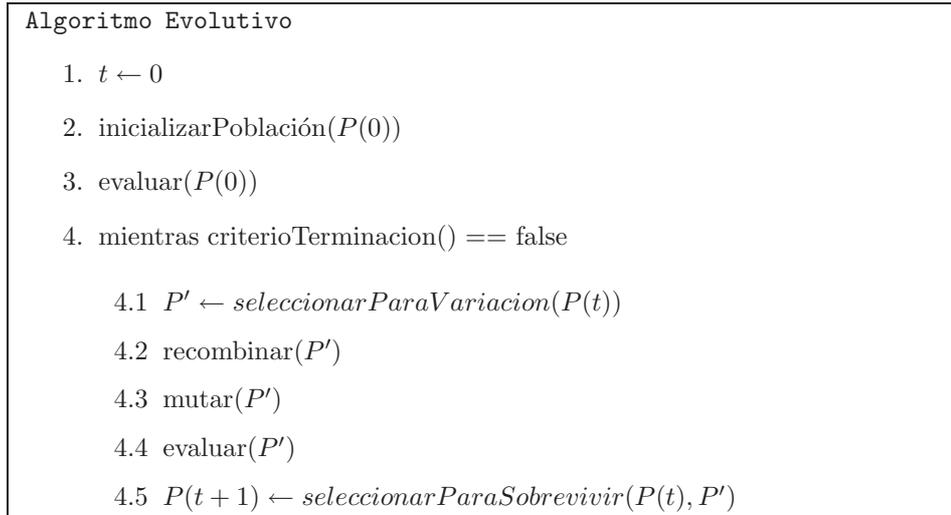


Figura 1.2: Esquema de un AE

Como hemos comentado, inspirados en estos principios de la evolución biológica, surgen los AEs. Tienen una población de individuos, los cuales, empleando operadores probabilísticos de mutación, selección y (en ocasiones) recombinación hacen que la población evolucione hasta conseguir soluciones con un menor valor asociado de la función objetivo (función del problema a optimizar).

Existen diferentes tipos de AEs: Los algoritmos genéticos [Gol89], programación genética [Koz92], estrategias de evolución [Bey01], y programación evolutiva [LP02]. Todos los AEs se caracterizan en que se componen de una población inicial (usualmente generada aleatoriamente) que sufre un proceso iterativo que conduce a la evolución de dicha población (a semejanza de la natural) mediante los siguientes procesos:

1. Se aplica un operador de selección que determina la probabilidad de cada individuo de perdurar en la generación siguiente, construyéndose así una población temporal.
2. Se aplica los operadores de mutación y recombinación a parte o la totalidad de la población temporal, con lo que se produce un conjunto de nuevas soluciones.

3. Se calcula el valor de la función de optimización de las nuevas soluciones generadas.
4. Se obtiene una nueva población a partir de la población temporal y los nuevos individuos generados.

Dichos pasos son realizados hasta que se cumple un criterio de terminación (se alcanza un límite máximo temporal o un número máximo de evaluaciones de la función objetivo). Este proceso está reflejado en el esquema de la Figura 1.2.

Este procedimiento de ir creando nuevos individuos a partir de los actuales individuos de la población, y en el que mejores individuos sustituyen a éstos es lo que produce lo que se denomina evolución de la población. El construir los nuevos individuos combinando características de buenas soluciones encontradas hasta el momento (almacenadas en la población) es lo que permite orientar la búsqueda hacia las regiones más prometedoras.

Todos los AEs comparten los mismos conceptos básicos aunque difieren en el mecanismo empleado para codificar las soluciones y los operadores que emplean para producir la siguiente generación.

Dentro de los AE destacan los algoritmos genéticos (AGs) [Gol89]. Los AGs son un tipo de AE caracterizados en que el principal mecanismo empleado para generar nuevos individuos es mediante un de recombinación (denominado cruce) que combina información de diferentes individuos para producir los nuevos individuos. Dentro de los AGs, en operador de mutación es empleado, por contra, con una probabilidad baja, simplemente como un operador capaz de aumentar la diversidad en la población. Los AEs como los AGs presentan la ventaja de ser algoritmos de búsqueda capaces de identificar regiones con alto rendimiento en dominios de búsqueda amplios y complejos, pero presentan, por contra, problemas para obtener soluciones precisas [KPTP01].

Con el paso del tiempo, otros modelos de AEs han sido propuestos. A continuación comentamos algunos de los nuevos modelos surgidos, y que consideramos de especial interés:

- La Búsqueda Dispersa [LM03]. En este modelo se mantienen dos subpoblaciones, una élite, de las mejores soluciones encontradas (para explotar las soluciones encontradas), y otra formada por individuos alejados de los individuos de la población anterior (para mantener cierta diversidad en el proceso de combinación). Además, es un modelo que considera el uso de

un proceso de mejora local, como una BL, para explotar más las soluciones encontradas. En la Sección 2.2.2 se hace una descripción más detallada.

- La evolución diferencial (o *Differential Evolution*, DE) [SP97, PRL05]. Este modelo se caracteriza en que, dado que las soluciones se representan mediante vectores de valores reales, las poblaciones de soluciones son modificados de forma sucesiva mediante operadores de suma y resta vectorial, así como operaciones de intercambios de valores, hasta que la población converge.
- Sistemas de Partículas (*Particle Swarm Optimization*, PSO) [KE95]. Los PSOs se caracterizan en que son modelos inspirados en modelos naturales de movimiento como las bandadas de pájaros. En estos, la población está compuesta de una serie de partículas que representan las soluciones, y la evolución de la población se realiza debido al desplazamiento en el espacio de búsqueda de dichas partículas. Dicho desplazamiento está principalmente formado por dos direcciones: Una hacia la mejor solución global encontrada; y la otra hacia la mejor solución encontrada por dicha partícula.
- Algoritmos Basados en estimación de Distribuciones (*Estimation of Distribution Algorithms*, EDA) [LLIB06]. Los EDAs se caracterizan en que no utilizan ni los operadores de cruce ni los de mutación para generar nuevas soluciones. Se genera la nueva población mediante muestreo de una distribución de probabilidad, la cual es estimada a partir de los mejores individuos de las generaciones anteriores. Un ejemplo de este tipo de algoritmos es el *Covariance Matrix Adaptation*, empleado en el Capítulo 4, y descrito con detalle en la Sección 4.1.1.

### 1.3. Algoritmos Meméticos

Se ha demostrado que para poder obtener buenos resultados es necesario que el algoritmo empleado utilice un buen equilibrio entre los principales factores durante el proceso de búsqueda: la exploración del espacio de búsqueda, y la explotación de las soluciones ya encontradas [ES98].

Los AMs son algoritmos especialmente indicado para obtener dicho equilibrio mediante el empleo de dos componentes distintos (AE y BL), cada uno de ellos especialmente orientado hacia cada uno de dichos elementos, permitiendo una mayor flexibilidad que permita ajustar ambos factores.

En esta sección vamos a introducir los AMs. En una primera parte, definiremos los AMs y sus componentes, y, posteriormente, presentamos el enfoque en el están inspirados.

### 1.3.1. Introducción a los Algoritmos Meméticos

Cada una de las técnicas anteriores (BL y AEs) presentan un especial énfasis en uno de estos componentes de la búsqueda. Las principales diferencias, que suponen tanto ventajas como desventajas, entre los método de BL y los AEs se deben a que los primeros potencian en su búsqueda la explotación de las mejores soluciones (ya que la nuevas soluciones son variaciones de la mejor encontrada hasta el momento), mientras que los AEs potencian más el factor de exploración (las nuevas soluciones al generarse mediante combinación de más de un individuo, generan soluciones distintas a cualquiera de sus progenitores). Por tanto, el proceso de exploración de vecinos de la BL permite obtener soluciones más precisas de forma muy eficiente (mediante una alta explotación de las soluciones encontradas); mientras que los AEs como los AGs, al mantener simultáneamente una población de múltiples soluciones, pueden realizar una exploración simultanea de varias regiones, pudiendo obtener una amplia búsqueda en dominios complejos.

El uso combinado de un AE junto con una BL posee propiedades beneficiosas al proceso de búsqueda, y ha sido presentado en la literatura [Mic96] bajo distintos nombres (algoritmos híbridos, algoritmos genéticos locales, etc.).

La incorporación de la BL permite abordar la necesidad de precisión durante todo el proceso. A diferencia de un AE, en el que el ajuste fino de las soluciones se produce sólo en las etapas finales, la BL permite aplicarlo durante todo el proceso de búsqueda.

Dentro de la gran variedad de algoritmos híbridos existentes [Tal02] existe una variedad denominada algoritmos meméticos (AMs) [Mos89, Mos99]. Existen distintas definiciones de lo que constituye un AM, con diferencias entre sí que han conducido (conforme aumentaba la variedad de hibridaciones) a distintas consideraciones sobre si un mismo algoritmo era un AM o no.

La definición inicial de Moscato [Mos89] lo define de la siguiente forma:

*Algoritmo Memético es una población de agentes que alternan periodos de auto-mejora (mediante búsqueda local) con periodos de cooperación (mediante recombinación) y competición (mediante selección).*

Hay varios aspectos importantes a tener en cuenta en esta definición:

- Se utiliza el concepto de agentes en lugar de individuos ya que se considera a los primeros una extensión de los segundos.
- Permite la hibridación con todo AE y proceso de búsqueda local.
- Tanto la selección como el reemplazo (*competición*) son procesos puramente competitivos.
- La reproducción es la encargada de crear nuevos agentes (mediante *cooperación* de los individuos existentes). Aunque puede aplicarse una gran variedad de operadores de reproducción, existen básicamente dos: Recombinación y Mutación.

Una pega que presenta la definición anterior es que no ofrece un criterio claro que permita distinguir, en un mundo de hibridación cada vez más creciente, lo que puede o no considerarse un AM. Una definición más precisa, práctica en este sentido y aceptada en la literatura sería la empleada por Krasnogor[KS05]:

*Un Algoritmo Memético es un Algoritmo Evolutivo que incluye una o más fases de búsqueda local en su ciclo evolutivo.*

Durante la tesis consideraremos esta definición.

Como aspecto esencial en esta definición está la consideración realizada acerca del *ciclo evolutivo*. Es decir, no se consideraría como AM un AE híbrido en el que la BL se aplicase sólo antes y/o después del proceso evolutivo. De esta forma, se podría decir que el carácter identificativo de los AMs dentro de los algoritmos híbridos es el de combinar etapas de AE y de BL dentro del mismo proceso evolutivo. El AE se encarga de realizar el proceso global de búsqueda mientras que la BL se aplicaría sobre las nuevas soluciones para mejorar la precisión de las soluciones encontradas.

El objetivo de los AMs es el de hacer que ambos componentes trabajen de forma cooperativa para conseguir una continua sinergia entre ambos que permita mejorar el proceso de búsqueda [Har94].

La Figura 1.3 muestra un posible esquema de un AM.

Los AMs se componen, por tanto, principalmente de:

**Algoritmo Memético**

1.  $t \leftarrow 0$
2. Generar la población inicial  $P(0)$ 
  - 2.1. Inicializar población inicial  $P(0)$  aleatoriamente
  - 2.2. Seleccionar un conjunto  $C \in P(0)$  de soluciones iniciales sobre los que aplicar la BL
  - 2.3. Aplicar la Búsqueda Local a los individuos  $i \in C$
3. Evaluar la población inicial
4. Mientras no se cumpla el criterio de terminación
  - 4.1. Seleccionar un conjunto  $P_{cruce}(t)$  de individuos a cruzar, considerando su *fitness*
  - 4.2. Aplicar operador de cruce sobre  $P_{cruce}$  y almacenar los descendientes en  $P'(t)$
  - 4.3. Opcionalmente mutar ciertos individuos del conjunto  $P'(t)$
  - 4.4. Aplicar a cada nuevo individuo un proceso de BL con probabilidad  $P_{BL}$
  - 4.5. Calcular el *fitness* de  $P'(t)$
  - 4.6. Generar  $P(t+1)$  a partir de  $P(t)$  y  $P'(t)$
  - 4.7.  $t \leftarrow t + 1$
5. Devolver individuo  $p$  de la población con mejor valor *fitness*

Figura 1.3: Esquema de un AM

- AE: responsable de la búsqueda global. Este algoritmo es el responsable de realizar la exploración y evitar que se produzca una convergencia prematura [ES91]. La labor de este componente es la de generar soluciones en las regiones más prometedoras. Debido a que el AE delega la función de aumentar la precisión de las soluciones obtenidas en el método de BL, el algoritmo puede centrarse en realizar una búsqueda con mayor diversidad, que contrarreste el aumento de presión selectiva debida a la aplicación de la BL. Este aspecto lo comentaremos con mayor detalle en la Sección 1.4.
- BL: responsable de la explotación de las soluciones más prometedoras.  
En el caso de optimización continua, existe un catálogo amplio de métodos de BL [Sch81]: Simplex, Solis Wets, Quasi-Newton, entre otros. En la Sección 1.5 comentamos este componente con mayor detalle.

Denominamos mecanismo de hibridación a la forma de combinar ambos componentes, es decir, cuándo aplicar la BL, a qué soluciones, con qué intensidad y la ratio de esfuerzo invertido en Búsqueda Local respecto al total, entre otros aspectos.

Es conveniente utilizar un mecanismo de hibridación que permita que ambos componentes puedan trabajar de forma complementaria. Si no se hace así, no hay garantías de que la incorporación de la BL mejore los resultados [Lan98] o bien, que dicha mejora no se deba a aplicarse dentro del propio el ciclo evolutivo [CE05].

En la Sección 1.6 hablaremos del mecanismo de hibridación con mayor detalle.

### 1.3.2. Enfoques de los Algoritmos Meméticos

De igual forma que los AEs están inspirados en la evolución natural [Dar59], los AMs están inspirados en un modelo natural [Dar76]. Darwin propuso en su modelo de evolución un sistema en el que la población mejoraba mediante la generación de nuevos individuos y un proceso competitivo en el que se propagan en el tiempo las características de los mejores adaptados al medio.

Se propusieron dos enfoques distintos para justificar la adaptación de las especies al medio:

- Lamarck propuso que, durante su ciclo de vida, los individuos realizaban

actividades que suponían cambios en su cuerpo que le permitían adaptarse mejor al medio, y que dichas mejoras eran transmitidas a su descendencia.

De esa forma, la actividad de los individuos con vista a adaptarse al medio orientaría la evolución de la especie a largo plazo.

- Darwin, por contra, proponía un proceso de evolución en el que las actividades de los individuos no son heredadas por sus descendientes. Desde el punto de vista evolutivo, los individuos, una vez generados, no cambian, y toda evolución de las especies es debida a pequeños cambios entre una generación y la siguiente, y un proceso competitivo en el que los mejores adaptados tendrán mayores posibilidades de obtener descendencia que los peores.

De esa forma, al heredarse más las características de los individuos más adaptados, evolucionaría la especie para adaptarse mejor al medio.

Desde la propuesta inicial, los descubrimientos de los mecanismos que almacenan y propagan las características propias de los individuos, los genes, han confirmado el enfoque Darwiniano frente al enfoque Lamarkiano [Dar76], demostrándose que no poseía este último base científica.

Por tanto, el decir que un AM sigue un enfoque lamarkiano es correcto en tanto implica un hecho cierto, que los individuos pueden ser modificados durante el proceso evolutivo, pero puede confundir ya que nos remite a un modelo natural (genético) que no es el propio del AM, sino que es el siguiente:

En [Dar76] se afirma que aunque parte del comportamiento de los individuos está programado en los genes, otra parte depende de comportamiento aprendido de sus progenitores, y se propone el concepto de *meme* como

*toda idea, melodía, creencia definida, moda de vestimenta [...] que en gran número constituyen el bagaje cultural de los individuos.*

Es decir, un meme sería todo elemento cultural. Dichos memes son aceptados por los individuos y determinan parte de su comportamiento.

En la misma fuente se detalla cómo estos memes son transmitidos:

*Del mismo modo que los genes se propagan en el acervo genético a través de gametos, los memes se propagan en el acervo memético saltando de cerebro en cerebro en un proceso que, en un amplio sentido, puede llamarse imitación.*

Esta cita señala, tal y como comentan Moscato y Cotta [MP03], la idea central de los AMs:

*mejoras individuales de las soluciones en cada uno de los agentes junto con procesos de cooperación y competiciones de tipo poblacional.*

Por tanto, los memes perduran entre los individuos mediante un proceso de imitación, que puede sufrir una evolución dentro de una población de forma simultánea a una evolución genética. La evolución cultural se caracteriza en que, debido a la amplia diversidad de memes que pueden ser aceptados por individuos dentro de la misma sociedad, un cambio en el grado de aceptación de uno o varios memes puede originar una modificación muy significativa del comportamiento de sus individuos en un plazo de tiempo muy corto. Esta ventaja competitiva de mayor velocidad de adaptación en el comportamiento aprendido se denomina efecto *efecto Balduwiniano* [Bal96], que no tiene nada que ver con un enfoque lamarkiano [Tur96].

Un claro ejemplo de esta ventaja es la historia de la evolución humana. Frente a grupos animales que poseen un comportamiento determinado principalmente por sus genes (y, por tanto, la posibilidad de adaptación es muy lento), en la especie humana cada individuo está determinado en mayor parte por un comportamiento aprendido dentro de la sociedad de individuos en la que se desarrolla. La mayor velocidad de adaptación social ha permitido a la especie humana una gran adaptación a entornos hostiles en un periodo muy corto (se colonizó el mundo entero sin prácticamente adaptación genética).

En los AMs, el comportamiento heredado está representado por el AE, mientras que el comportamiento aprendido que permite mejorar al individuo está representado por la BL. El efecto Balduwiano de los AMs puede observarse en su mayor velocidad de adaptarse a los problemas (mayor velocidad en la obtención de soluciones precisas).

## 1.4. Búsqueda Global aportada por los Algoritmos Evolutivos

Como ya hemos comentado, el AM requiere de un componente responsable principalmente de realizar la búsqueda global. El objetivo de éste es explorar el

espacio de búsqueda, orientando la búsqueda hacia las regiones más prometedoras, tratando al mismo tiempo de no quedar atrapada en óptimos locales.

Un AE adecuado para formar parte de un AM debe de poseer las siguientes características:

- Que introduzca diversidad para favorecer la exploración del espacio de búsqueda. Existe, asociado a una BL un concepto muy importante, denominado *base de atracción*. Se denomina base de atracción de un mínimo local al conjunto de soluciones para las cuales el proceso de BL conduce a dicho óptimo local. Para que el uso del AE pueda servir para evitar quedar atrapados en óptimos locales es necesario que las soluciones del AE que sean mejoradas pertenezcan a distintas bases de atracción, y no se produzca convergencia prematura.
- Que sea capaz de aceptar las nuevas soluciones por el sistema de BL en la población. Eso supone, por un lado, que sea receptivo a esas soluciones (para que la BL pueda aplicarse de forma efectiva), pero que al mismo tiempo mantenga la diversidad (los resultados de varias BL pueden conducir a un mismo óptimo local, lo cual podría reducir la diversidad de la población si el mecanismo de introducción de nuevas soluciones no lo evitase).
- Que el incorporar las nuevas soluciones precisas permita orientar mejor la búsqueda. Es decir, que el proceso de búsqueda sea susceptible (en un cierto grado) a una mejora de una parte de la población.

Existe un gran número de esquemas evolutivos (Algoritmos Genéticos, AG [Gol89], *CHC* [Esh91], Búsqueda Dispersa (BD) [Glo77, LM03], ...). De entre todos ellos, interesaría estudiar qué AEs presentan un mejor comportamiento para diseñar un AM que consiga soluciones de gran calidad, para lo cual un estudio experimental se hace necesario para realizar una afirmación en un sentido o en el otro. En el Capítulo 2 se presenta un estudio comparativo en este sentido.

## 1.5. Búsqueda Local y Explotación

A diferencia del componente anterior, la BL tiene como objetivo explotar las soluciones más prometedoras.

La BL está definida por los siguientes elementos:

- **Función de transformación:** Permite generar una nueva solución a partir de una solución actual. Dicha función de transformación suele ser no determinista, por lo que a partir de una misma solución, podrían generarse distintas soluciones.
- **El número de soluciones generadas a partir de una misma solución actual,** de las que se escogerá la nueva siguiente solución. A este número se le denomina *anchura de las búsqueda local*.
- **Criterio de selección de la siguiente solución:** En general, es una práctica común escoger de entre las generadas y la solución actual aquella con mejor valor de la función objetivo (esquema de ascensión de colinas), aunque ciertos algoritmos como el Enfriamiento Simulado permiten que una solución sea sustituida por otra con peor valor de la función objetivo [KGV83].
- **Criterio de parada:** En optimización continua, dado que no se genera todo el vecindario, sino un subconjunto del mismo (indicado por la anchura), en ningún caso se puede saber si la solución actual es un óptimo local o no [Har94], por lo que un criterio factible de parada es un tiempo máximo o un número máximo de iteraciones, denominado *profundidad de la búsqueda local*, o número total de evaluaciones, denominado en ese caso *intensidad de la búsqueda local*.

En optimización continua existe un amplio catálogo de métodos de BL sobre búsqueda continua: métodos del Gradiente, Simplex, Solis Wets, Quasi-Newton, ... [Sch81]. Nos interesa poder utilizar ese amplio catálogo de métodos de BL en nuestros AMs con codificación real. La gran diversidad de métodos de BL supone una gran diferencia entre sus comportamientos, lo cual implica que los resultados finales puede depender fuertemente de la elección de una u otra BL.

A menudo se llega a construir un AM mediante la incorporación de una BL con información del problema, por lo que la incorporación de la BL suele suponer una mejora. Sin embargo, a veces, si la BL utiliza conocimiento avanzado del problema, puede darse el caso de que la bondad se deba casi en exclusiva a la BL y no al uso de un componente evolutivo [CE05].

La elección del método de BL adecuado es un problema complejo, ya que depende del problema, y también puede depender del estado actual de la búsqueda.

Esta dependencia, que imposibilita la elección de un único método de BL como el más adecuado, ha sido tratado por Soon Ong y Keane [OK04], que proponen como alternativa un conjunto de métodos de BL y un mecanismo que permite elegir para cada individuo la BL a aplicar mediante el empleo de distribuciones de probabilidad. Inicialmente, los métodos considerados tienen igual probabilidad de ser aplicados, pero a medida que se reitera la aplicación de éstos, aquellos métodos que obtienen mejores resultados (mayor mejora) aumentarán su probabilidad asociada.

El esquema anterior ofrece una respuesta al problema de la elección de la Búsqueda Local adecuada, mediante un conjunto de métodos de BL y un proceso adaptativo que permite aplicar en cada caso el más adecuado. La existencia de este trabajo, que ofrece una alternativa que consideramos muy adecuada, ha hecho que esta problemática de elección (de la BL) no sea abordado por nuestra memoria, pudiéndonos centrar más en otros problemas, no menos importantes.

## **1.6. Mecanismo de Hibridación entre los Algoritmos Evolutivos y la Búsqueda Local**

Debido al aumento en coste computacional que supone la incorporación de la BL, es necesario elegir cuidadosamente tanto aquellas soluciones sobre las que aplicarla, como el número máximo de evaluaciones durante cada BL. Ambas decisiones determinan de forma importante la posibilidad de alcanzar un equilibrio entre los factores de eficiencia y mejora, para que la mejora obtenida compense la merma de evaluaciones en la búsqueda global. Si no se hace así, no hay garantías de que la incorporación de la BL mejore los resultados [Lan98].

Para eso es necesario asegurarse que los resultados obtenidos por la combinación/sinergia de los dos componentes anteriores (AEs y BL) son mejores que los obtenidos por cualquiera de éstos de forma separada: Que la incorporación de la BL permite mejorar la explotación de las soluciones encontradas (siempre que dicha mejora compense la disminución de exploración), y que el uso de un AE permita aplicar la BL sobre soluciones separadas entre sí, con lo que se reduzca el problema de la convergencia prematura.

Es decir, el buen comportamiento no depende únicamente (ni siquiera principalmente) del conjunto del AE-BL empleado, sino también de aplicar un adecuado esquema de hibridación que permita a los componentes trabajar cooperativamen-

te [Tal02, OLZW06, KS05]. Denominamos mecanismo de hibridación al esquema que determina el modo en que son combinados las etapas del AEs y BL. Hart [Har94] afirma que un modelo de hibridación debe de tratar los siguientes factores:

- Frecuencia de aplicación de la BL: Con qué probabilidad se aplica la BL a los nuevos individuos.

El modelo inicial de AMs aplicaban la BL con una cierta intensidad a toda nueva solución generada por el AE. Este modelo, que sigue vigente en ciertos algoritmos como la Búsqueda Dispersa [Glo77, LM03] posee la ventaja de realizar un ajuste fino de toda nueva solución generada.

Sin embargo, esto supone un coste excesivo ya que por cada nueva solución generada por el AE se aplica una BL durante  $I_{BL}$  iteraciones por lo que el ratio de esfuerzo invertido en la BL es de  $\frac{I_{BL}}{I_{BL}+1}$ , que podría ser prohibitivo en muchos problemas. En esos caso a veces se acaba reduciendo el tamaño de la población para poder aplicar el AE durante varias iteraciones, con lo que se reducen las característica exploratorias del AM. Una alternativa a este inconveniente es aplicar la BL únicamente a un porcentaje reducido de las soluciones [Har94, IM03].

- Conjunto de individuos sobre los que aplicar la BL.

Si la BL no se aplica a todos los individuos de la población, es necesario seleccionar el conjunto de individuos a mejorar.

El objetivo de un criterio de selección adecuado es doble:

- Aplicar la BL sobre soluciones potenciales prometedoras.
- Evitar aplicar la BL sobre soluciones próximas, ya que no aportaría un beneficio significativo.

Un criterio muy empleado es considerar cierto valor de probabilidad de aplicación de la BL ( $P_{BL}$ ) independiente de las características de las soluciones. El procedimiento es sencillo, para cada nueva solución se genera aleatoriamente un número  $u \in [0, 1]$ , y se aplica la BL sobre dicho individuo si y sólo si  $u < P_{BL}$ . Hart propone una probabilidad  $P_{BL}$  baja, de 0.0625.

Distintos autores [OLZW06, BBTZ04, Lan98] han propuesto AMs que adaptan el criterio de elección sobre si aplicar la BL a un individuo, en función del comportamiento de éste o de un conjunto de individuos.

En capítulos posteriores de esta memoria consideraremos la conveniencia de considerar las características de los individuos a la hora de considerar si aplicar la BL sobre ellos o no.

- La intensidad en cada aplicación de la BL.

A la hora de aplicar la BL es conveniente alcanzar un equilibrio entre el porcentaje de individuos sobre los que aplicar la BL y la intensidad de su aplicación.

Si este valor es fijo (independiente de las características de los individuos) debe de mantenerse bajo, o bien el número de soluciones a mejorar, o bien la intensidad de cada aplicación. Es más, con una intensidad fija el grado de explotación de cada solución será el mismo, con lo que la mayor o menos explotación por parte de la BL de una determinada región estará determinada principalmente por el número de individuos de dicha región sobre los que se aplicará la BL. Así pues, una región será explotada más por la BL cuando se aplique más sobre individuos de dicha región, aumentándose así el riesgo de convergencia prematura. Por otro lado, el uso de una intensidad adaptativa permitiría aplicar una mayor profundidad sobre las regiones más prometedoras sin aumentar el riesgo de convergencia prematura.

En los últimos años se han presentado en la literatura algunos AMs que aplican una intensidad adaptativa [Smi03, MHL05a, BBTZ04, SEAL04].

Al esquema de combinación de la BL con el AE se le dedicará gran atención durante el Capítulo 4 de la tesis, por lo que no entraremos en más detalle en este apartado.

## 1.7. Repaso Histórico de los Algoritmos Meméticos

Desde muy temprano, ya en 1977, vienen presentándose AEs incorporando una BL para obtener resultados de mejor calidad, como la Búsqueda Dispersa o *Scatter Search* [Glo77] que definió un esquema concreto de hibridación [Glo98a] que sigue de plena actualidad [MLG06].

Estos modelos, al igual que otros modelos de evolutivos, inicialmente fueron diseñados para abordar problemas combinatorios, por lo que utilizaban una codificación binaria [Gol89, Esh91]. Posteriormente, se empezaron usar los mismos algoritmos en problemas de optimización continua [ECS93].

En 1989, Moscato [Mos89] presentó los algoritmos meméticos, acuñándose el término, como una nueva técnica para mejorar los procesos de búsqueda combinando un algoritmo exploratorio junto con un proceso de mejora individual de las soluciones.

En 1994, Hart [Har94] presentó en su tesis la conveniencia de uso de algoritmos híbridos. En esta tesis ya se empezaron a plantear importantes cuestiones, algunas de las cuales son:

- La conveniencia de la BL como medio de mejora de la precisión.
- Que el uso de una BL no implica necesariamente una mejora, que depende mucho del mecanismo de hibridación empleado.
- Que por el coste asociado a la BL no es aplicable la BL a toda nueva solución, y propone un modelo de hibridación usando una probabilidad de BL muy pequeña (0.0625 o inferior).
- Plantea que la BL puede aumentar el riesgo de convergencia prematura.

Desde mediados de los 90 hasta el año 2000 se produjo un esfuerzo en adaptar los componentes de los AEs para codificación real, en particular, el cruce. Este interés se afianzó debido a estudios como los de Herrera, Lozano y Verdegay [HLV98], que demostraron la conveniencia de utilizar, para abordar problemas de optimización continua, algoritmos que representasen las soluciones mediante valores reales, en vez de representar una solución real mediante valores discretos o binarios. De esta forma, al acercar el dominio de exploración al dominio del problema, se permite incorporar más fácilmente información del problema en la exploración. En [HLS03] puede observarse una taxonomía de operadores de cruce diseñados para codificación real.

En 1997 se presentó el teorema del *Non-free lunch* para optimización [WM97], que mantiene que todo algoritmo de optimización que aborde problemas sin información aplicando un mismo proceso iterativo (como la práctica totalidad de los AEs no adaptativos) son equivalente: que sin usar información sólo se puede afirmar que un algoritmo es mejor que otros para ciertas funciones, y no que sea mejor en general. Este teorema, de graves consecuencias en el campo de optimización, ha supuesto un importante acicate para la incorporación de conocimiento del problema en los algoritmos que lo abordan. Esto ha aumentado el interés sobre los AMs, dado que en éstos es muy sencillo incorporar conocimiento del problema.

En este ambiente de creciente interés por los AMs surgen otros estudios en profundidad continuando con la línea de investigación marcada por Hart, como el de Land en 1998 [Lan98]. Aunque centrado en la problemática combinatoria, plantea cuestiones importantes también en optimización continua. Destacamos dos:

- Destaca la importancia de la intensidad de la BL. El empleo de una alta intensidad de BL podría mejorar considerablemente las soluciones encontradas, pero reducir demasiado la diversidad. Una baja intensidad mantendría mayor diversidad, pero podría ofrecer poca mejora respecto a un modelo no memético (sin BL).
- Propone la conveniencia de buscar métodos que permite obtener una intensidad adecuada para cada BL.

A finales de los años 90s el concepto de algoritmo memético empezó a popularizarse [Mos99, MF99].

En el año 1999 Sinha [Sin99], y en el año 2000 Merz [Mer00], estudian por separado los AMs desde una perspectiva constructiva: Qué factores, o criterios de diseño, de un AM son deseables para poder obtener el mayor provecho posible de la nueva estructura (AE + BL). Sin embargo, mientras que Sinha se centra en buscar criterios globales de diseño, Merz aborda distintos problemas muy conocidos dentro de la optimización combinatoria e intenta abordarlos mediante un AM. Como fruto de este estudio, Merz consiguió que distintos AMs constituyan actualmente el estado del arte para determinados problemas en el entorno discreto, como en el caso del QAP [MF00]. El éxito de Merz se apoya en una cuidadosa construcción del AM para cada problema, con una BL específica para el mismo.

En el año 2000, Krasnogor [Kra02] aborda la problemática de la adecuada elección de la BL dentro de los AMs. Una de las principales propuestas de Krasnogor es el de representar dentro del cromosoma las características de la BL. La idea es, recuperando en parte el modelo natural en el que está basado 1.3.2, permitir que la BL evolucione de forma conjunta con los individuos. El procedimiento es sencillo: una vez representada la BL dentro de los cromosomas, definir métodos de cruce y mutación que actúen sobre dicha representación y permitan surgir nuevos métodos de BL.

En el 2003 se publicaron artículo de Moscato y Cotta [MC03, MP03], que sirvieron para dar más a conocer los conceptos de los AMs al público, tanto en un ámbito general como en un ámbito latinoamericano, respectivamente.

Para abordar la problemática de la correcta elección de la BL, Ong y Keane [OK04] proponen en el 2004 no aplicar una única BL, sino considerar una batería de métodos de BL. Así, siguiendo un esquema adaptativo, se podría aplicar para cada función (y en cada estado de la BL) más aquella o aquellas BLs que ofrezcan una mayor mejora. No entramos más en detalle, porque ya se ha comentado en detalle en 1.5.

En el 2004, se publicó el libro de Hart, Krasnogor y Smith, *Recent Advances in Memetic Algorithms* [HKS04], que supuso el primer libro centrado totalmente en los AMs. En él, se recogen tanto aspectos metodológicos (robustez, criterios de diseño, . . .), como aplicaciones de los AMs.

El campo de los AMs está vigente y activo (como se puede observar de las taxonomías presentadas por Talbi [Tal02], y Krasnogor y Smith [KS05]). Por otro lado, una nueva variante o extensión de los AMs son los AM adaptativos, los cuales permiten abordar aspectos de la hibridación mediante el empleo de parámetros adaptativos o *autoadaptativos* [BD01]. Es tal el área de interés que estos AMs genera que ya se ha producido una amplia variedad de éstos (en [OLZW06] se presenta una taxonomía específica de AMs adaptativos).

Un hito interesante fue la Sesión Especial de Optimización Real del *2005 IEEE Congress on Evolutionary Computation*. Es destacable el gran número de algoritmos híbridos. 6 de las 11 propuestas aplican una BL: 2 de ellas con un esquema multiarranque [AH05b, AH05a], 3 AMs [MHL05a, LS05, QS05], y un algoritmo que aplica un proceso de BL al final del algoritmo [GML05]. Además, se fijaron, por primera vez, unos criterios comunes de experimentación que hicieron posible comparar las distintas propuestas entre sí.

Como ejemplos de la buena salud e importancia de los algoritmos híbridos y meméticos, se ha definido una Sesión Especial de Algoritmos Meméticos en el *2007 IEEE Congress on Evolutionary Computation*, y se han planteado dos números especiales de Algoritmos Meméticos en el *IEEE Transactions on Systems, Man and Cybernetics*: Uno en el 2005, y otro en el 2007.

Por último, no puedo terminar este repaso sin resaltar el gran trabajo de Ishibuchi en el campo de la problemática multiobjetivo. A través de una serie de investigaciones a partir del 2001 [IYM02, IM03, IY03, IN04] ha presentado un modelo de AMCR para problemas multiobjetivos, el MOGA, de obligada referencia por sus excelentes resultados en el campo multiobjetivo.

## 1.8. Ventajas de los Algoritmos Meméticos

Se ha comprobado que la utilización de AMs posee varias ventajas en los problemas de optimización continua:

- Permite establecer de forma explícita (y ajustables) criterios que equilibren exploración y explotación. Al definirse cada componente orientado a fomentar únicamente uno de los objetivos, y delegar la responsabilidad de un adecuado equilibrio entre ambos en el modelo de hibridación, se facilita el establecer mecanismos para equilibren el empleo de ambos factores.
- Se alimentan de las investigaciones en otros campos, al poderse emplear distintos AEs o BLs ya propuestos.
- Incorporación de información específica del problema: Su estructura facilita el uso de información sobre los problemas para obtener buenos resultados (principalmente en el método de BL, aunque también puede darse en el AE, como el operador de cruce).

En problemas reales es posible utilizar en los componentes información sobre dichos problemas para poder obtener buenos resultados [dSCCK02, MBM05]. Este enfoque ha permitido que en ciertos problemas (QAP, ...) los AMs constituyan el estado del arte [MF00, MK04].

Esta misma posibilidad se puede dar también, aunque en menor grado, no sólo sobre problemas concretos, sino sobre categorías de problemas: Problemas multimodales [IN04, IM03], problemas de combinatoria [MF00, KS00], ...

- Incremento de la velocidad de convergencia:  
Mientras que en un AE la explotación es realizada principalmente durante las últimas generaciones, en un AM se realiza durante todo el proceso (mediante la aplicación de la BL). Por tanto, el AM puede alcanzar el mismo grado de precisión más rápidamente, con lo que el AM puede alcanzar en menos evaluaciones resultados equivalentes a los obtenidos por el AE.
- El AM puede alcanzar mejores resultados que el AE:  
Los AEs suelen presentar problemas de precisión. El uso de la BL permite mejorar rápidamente la precisión, con lo que se introducirían mejores soluciones en la población, que pueden ser utilizadas por el AG para obtener

aún mejores (utilizando las anteriores). Además, el utilizar dentro del AE soluciones mejoradas previamente por la BL puede ayudar a la búsqueda ya que dichas soluciones son más representativas de su vecindario (en un AE una mala solución dentro de una buena región no atraería la búsqueda, pero en el AM esta misma solución podría ser rápidamente mejorada, orientando la búsqueda hacia dicha región prometedora) [Har94].

## Capítulo 2

# Análisis de Algoritmos Evolutivos Híbridos

Como se se indicó en el planteamiento de la memoria, el objetivo de la misma es el diseño de AMCRs. Esto implica un diseño cuidadoso de sus principales elementos: El AE encargado de la búsqueda global, el método de BL, y el mecanismo de hibridación empleado para combinarlos. En este primer capítulo de estudio y diseño, nos centraremos en el primer componente, el AE, del cual ya se comentó en el Capítulo anterior su importancia.

Dado que se han planteado en la literatura numerosos AMs para optimización continua [KS05], sería inadecuado abordar esta problemática sin considerar primero las tendencias marcadas por éstos. Hemos observado que, a diferencia de la BL, que suele ser específica del problema en cuestión que se está intentando abordar, no se suele utilizar modelos de AEs específicos para dichos problemas. Aunque es común utilizar operadores específicos del problema (como los operadores de cruce o mutación), el esquema general se suele elegir de entre la abundante bibliografía. Como en dicha bibliografía se propone una amplia variedad de esquemas de AEs (a menudo con distintas variantes), que presentan un comportamiento muy característico, el abanico de elección es muy amplio. Lamentablemente, no existen criterios consolidados sobre los que apoyar tan decisiva decisión. Podrían consultarse estudios comparativos de AEs y elegir aquél que demostrase un adecuado comportamiento. Sin embargo, la incorporación de una BL a un AE potencia el elemento de explotación, alterando el equilibrio ex-

ploración/explotación, con lo que se puede afectar de forma significativa el buen comportamiento del algoritmo. Por tanto, sería conveniente poder comparar el comportamiento de distintos AEs conforme se le incorpora un proceso de BL.

En este capítulo presentaremos un estudio de estas características, sobre un conjunto, pequeño, pero que creemos representativo, de distintos AEs y su evolución conforme se le añade/incrementa un procedimiento de BL. El objetivo es intentar determinar qué esquemas de AEs son más prometedores para usarse en la construcción de un AM.

Este capítulo presenta la siguiente estructura. En la Sección 2.1, mostramos distintas medidas utilizadas en la literatura para fomentar la diversidad de los Algoritmos Meméticos. En la Sección 2.2, presentamos los distintos AEs considerados en la comparación. En la Sección 2.3, se incluye el estudio comparativo. Finalmente, en la Sección 2.4, se presentan las conclusiones finales del capítulo.

## 2.1. Fomento de diversidad

La necesidad de modificar el AE de exploración para conseguir una mayor diversidad en los AMCRs siempre se trata, ya sea de forma directa o indirecta [Tal02],[OLZW06],[KS05]. Examinaremos algunos enfoques:

- En [MSB91] se propone como mecanismo de fomento de diversidad el empleo de AGs distribuidos, que mantienen en paralelo varias subpoblaciones, procesadas cada una por un AG de forma independiente. Su principal ventaja es la preservación de la diversidad mediante la separación en distintas subpoblaciones.
- Merz [Mer00],[MF00],[MK04],[Mer04] ha alcanzado AMs que abordan de forma muy eficiente problemas clásicos de optimización continua (viajante de comercio, QAP, ...). Sus buenos resultados son un modelo a seguir, y se apoya en:
  - Por un lado, el uso de operadores de cruce y mutación específicos del problema. En general, los operadores de cruce elegidos buscan mantener la diversidad durante la búsqueda. Además, el conocimiento del problema le permite el empleo de una BL capaz de explotar de forma exitosa en un entorno cercano (en el dominio del problema).

- Por el otro, no se ciñe a un esquema de AE rígido, sino que lo adapta en cada caso al problema considerado. Por ejemplo, suele incorporar mecanismos de reinicio. Conforme avanza el algoritmo, las soluciones contenidas en la población se acercan hasta estar concentradas dentro de una fracción muy pequeña del área de búsqueda. Se dice en ese caso que la búsqueda ha convergido. Al darse con mayor frecuencia esta situación en los AMs, las propuestas de Merz suele aplicar en ese caso un procedimiento de reinicio, que perturba a la población para que los individuos estén alejados entre sí. Es decir, dicho mecanismo representa un medio para escapar de las regiones subóptimas del espacio de búsqueda.
- Nagata y Kobayashi [NK97] describen un potente AM con un operador de cruce inteligente en el que el método de BL se integra dentro del operador genético. Es más, usa poblaciones varios órdenes de magnitud mayor que los empleados por otros autores, para incrementar la diversidad.
- Krasnogor y Smith [KS00] introducen un esquema de hibridación de un AM que utiliza para mejorar las soluciones encontradas información estadística sobre la población del AG. Su AM está compuesto de dos procesos de hibridación, un AG y un método de Montecarlo que realiza dos tareas: por un lado, cuando la población diverge, actúa como un procedimiento de BL; mientras que cuando la población converge, su objetivo es diversificar la búsqueda.
- Seront y Bersini [SB00] presentan un AM con un método de clustering que puede reducir el costo total de la BL evitando redescubrir varias veces los mismos mismos óptimos locales. Además, el método de clustering ofrece información que puede ser empleada para mantener la diversidad poblacional. Kemedede [Kem96] presentó un modelo de AM basado en estrategias de evolución que aplican ideas parecidas.
- Por último, Partasarathy, Goldberg y Burns [PGB01], para aplicar el uso de AMs en problemas multimodales emplean métodos de nichos adaptativos propuestos por Goldberg y Wans [GW97], para mantener de forma estable el nivel de diversidad de la población durante la búsqueda.

## 2.2. Modelos de Algoritmos Evolutivos para AMs

Dentro del amplio conjunto de AEs se ha determinado considerar un conjunto, pequeño, pero con la intención de que sean suficientemente representativos, de AEs utilizados en los AMs.

Pasamos a comentar de forma muy resumida los AEs considerados en nuestro estudio comparativo:

- Los Algoritmos Genéticos (AGs) [Gol89]. Son AEs que se caracterizan en que usan el operador de cruce como el principal mecanismo de generación de individuos, dejando el operador de mutación como un operador adicional para aumentar aleatoriamente la diversidad de la población.

Desde los comienzos de los AEs, los AGs han gozado de amplia popularidad debido a su robustez y capacidad de exploración de dominios complejos. Es, además, un esquema de algoritmo muy empleado en la construcción de AMs.

Hay varios modelos de AGs que consideraremos por separado:

- El modelo Generacional (AGG) [Gol89]. En este modelo se genera en cada generación una nueva población, compuesta de un conjunto de soluciones generadas por cruce de individuos de la población anterior, y de un subconjunto de la población anterior (elegidos mediante un operador de selección).

Los individuos que no han sido seleccionados para ser incluido en la nueva población son eliminados, a excepción de la mejor solución encontrada (*criterio elitista*).

- El modelo estacionario (AGE) [Sys89, Whi89]. En este modelo en cada generación se realiza un único cruce. Se selecciona los padres de dicho cruce, obteniendo únicamente una o dos soluciones hijas. Luego, se decide si el nuevo individuo debe de introducirse en la población, y en ese caso, cuál de las soluciones actuales debe de eliminarse para hacer espacio al nuevo individuo. Es un modelo que permite mucha flexibilidad, mediante una adecuada selección de los padres y de la estrategia de reemplazo.

Los AGs son uno de los AEs más antiguos, han sido ampliamente utilizados y siguen siendo muy populares en sus distintas presentaciones (binaria, dis-

creta y real). Son actualmente uno de los AEs más empleados por modelos recientes de hibridación [BBTZ04, CS03, OK04, Smi03].

- CHC [Esh91]. Este algoritmo, bastante semejante a los AGs anteriores, se caracteriza en dos aspectos:
  - Uso de una población elitista. En la población se guardan los *Popsize* individuos con mejor valor para la función objetivo de entre los generados hasta ese momento. Esto implica que se eliminará un individuo de la población únicamente para ser reemplazado por un otro con mejor valor.
  - Estrategias para fomentar la diversidad en la población: con el objeto de mantenerla, se establece un mecanismo para evitar cruzar entre sí individuos demasiados parecidos entre sí (denominado prevención de incesto [ES91]); además, para introducir diversidad en el caso de la población haya convergido prematuramente, cuenta con un mecanismo de reinicio.

Aunque inicialmente fue definido con codificación binaria y ha sido muy utilizado en problemas combinatorios por su sencillez y robustez, los propios autores presentaron poco después un modelo continuo que ha sido también ampliamente empleado en problemas reales [ECS93].

- Sistema de Partículas, (Particle System Swarm, PSO) [KE95]. Los PSOs se caracterizan en que son modelos inspirados en modelos naturales de movimiento como las bandadas de pájaros. Los PSOs son inicializados con soluciones aleatorias (denominadas partículas o *swarm*) y luego, en cada generación, cada individuo, o solución potencial, se desplaza por el espacio de búsqueda del problema con una velocidad propia, y que es adaptada dinámicamente en función de las experiencias obtenidas en su vuelo tanto por él mismo como por su grupo social.

El movimiento de cada individuo está influido por dos componentes: Una individual e independiente del resto (hacia la mejor posición encontrada por dicha partícula), y la dirección marcada por el individuo con mejor comportamiento (líder de la bandada).

Estos modelos han sido empleados inicialmente en sistemas de reconocimiento de formas, pero su ámbito de aplicación está siendo ampliado debido a sus buenos resultados. En concreto, últimamente han surgido varios

modelos de sistemas de partículas para dominios concretos, algunos de los cuales aplicando modelos meméticos [LCC<sup>+</sup>05, LS05, STCSQ07].

- La Búsqueda Dispersa (BD)[Glo77]. Este modelo de AE posee una particularidad importante respecto al resto. De los modelos comparados, es el único que, desde su concepción, se diseñó pensando en ser utilizado con un proceso de mejora local (como una BL). Aunque se ha empleado principalmente en problemas combinatorios se ha mostrado también como un algoritmo adecuado en codificación real [HLM06].

A continuación pasamos a describir con mayor detalle cada uno de los AEs utilizados en el estudio comparativo.

### 2.2.1. El algoritmo CHC

La idea clave de CHC [Esh91] es la de ofrecer un AE que combina una alta presión selectiva (sólo son mantenidos los mejores individuos en la población) con distintos mecanismos orientados a mantener una alta diversidad durante todo el proceso de la exploración. Esta combinación es la que caracteriza al algoritmo.

La alta presión selectiva es mantenida mediante una población elitista. En cada generación se forma una nueva población temporal, compuesta por los  $M$  miembros de la población actual junto con la población de descendientes obtenida de ésta. De este conjunto se seleccionan los mejores  $M$  individuos (con mejor valor para la función objetivo) para formar la nueva población. En el caso de que un individuo anterior y un nuevo individuo tengan igual valor de la función objetivo se preferirá al descendiente (para introducir mayor diversidad).

Los principales mecanismos utilizados para mantener la diversidad (y disminuir así el riesgo de convergencia prematura) son:

- Un cruce que introduce diversidad.

La propuesta binaria utiliza el cruce HUX, que genera una amplia diversidad (cambiando la mitad de los alelos que se diferencien entre padres, elegida esa mitad aleatoriamente del total). De esta forma se garantiza siempre que los descendientes presenten la máxima distancia de Hamming respecto a sus progenitores, introduciendo gran diversidad en la población.

En la variante continua se considera el operador de cruce  $BLX - \alpha$  como sustituto, ya que es un operador que permite obtener una amplia diversidad. En el apartado [ES93] se describe este operador con más detalle.

- Un mecanismo para mantener la diversidad.

A la hora de cruzar individuos, se selecciona para cada individuo aleatoriamente otro con el que emparejarlo. Sin embargo, no todas las parejas formadas originarán descendientes. Antes de realizar el cruce, CHC comprueba que la mitad de la distancia entre los padres sea mayor que un cierto valor umbral ( $d$ ), a este mecanismo se le denomina *prevención de incesto*. A la distancia  $d$ , se le asigna un valor inicial y dicho valor disminuye cada vez que en una generación el mecanismo de incesto haya impedido cruzar toda pareja seleccionada.

El efecto de este mecanismo es que se seleccionen únicamente los padres con mayor potencial de diversidad, disminuyéndose de forma automática la diversidad exigida cuando la población converge de forma natural.

En el modelo binario se aplica la distancia de Hamming para medir la distancia entre dos individuos. Para calcular el valor  $d$ , éste se inicia a  $L/4$  (siendo  $L$  la longitud del cromosoma), y cada vez que en una generación no se generan nuevos individuos (por no permitirse ningún cruce) disminuye en 1.

En el modelo continuo se desea mantener el mismo esquema en la medida de lo posible. Así pues, para poder seguir aplicando Hamming como medida de similaridad entre dos individuos, se hace necesario aplicarles antes un proceso de conversión binaria. El umbral de diferencia  $d$  es inicializado a  $D_{max}/4$ , donde  $D_{max}$  es la longitud del individuo aplicando dicha conversión binaria, y se puede calcular mediante la siguiente expresión:  $D_{max} = \sum_{i=1}^N param_i \cdot nbits_i$ , donde  $N$  es el número de parámetros estimados en nuestro problema y  $nbits_i$  es el número de bit necesarios para codificar el  $i$ -ésimo parámetro. Al igual que en el modelo discreto, por cada generación sin descendientes este valor disminuye una unidad.

- Un proceso de reinicio que sustituye al operador de mutación de los AGs.

A diferencia del resto de mecanismos empleados, éste es un proceso que introduce de forma directa diversidad en la población, y es aplicado únicamente cuando detecta que la población ha convergido.

Para determinar cuando la población converge, utiliza el valor umbral  $d$ . Cuando  $d < 0$  aplica un reinicio de la población, manteniendo únicamente al mejor individuo encontrado hasta el momento (manteniendo así el elitismo en la población).

La Figura 2.1 muestra el esquema general del algoritmo CHC.

#### Algoritmo CHC

1. inicializarPoblación( $P(0)$ )
  2. evaluar( $P(0)$ )
  3.  $d \leftarrow L/4$
  4. mientras criterioTerminacion() == false
    - 4.1 Agrupar aleatoriamente la población  $P(t)$  en parejas  $(I_i, I_j)$
    - 4.2  $Offsprings \leftarrow \phi$
    - 4.3 Para cada par  $(I_i, I_j)$ 
      - 4.3.1  $distancia \leftarrow Distancia(I_i, I_j)$
      - 4.3.2 sólo si ( $distancia/2 < d$ )
        - ◇  $NewIndividual \leftarrow Crossover(I_i, I_j)$
        - ◇ (Opcional) Si se considera adecuado aplicar proceso de mejora sobre  $NewIndividual$
        - ◇  $Offsprings \leftarrow Offsprings + NewIndividual$
    - 4.4  $P'(t+1) \leftarrow P(t) + Offsprings$
    - 4.5  $P(t+1) \leftarrow M$  mejores individuos  $P'(t+1)$
    - 4.6 Si ( $Offsprings == \phi$ )  $d \leftarrow d - 1$
    - 4.7 Si ( $d < 0$ )  $ReiniciaPoblacion(P(t+1))$ ,  $d \leftarrow L/4$
- Devolver mejor individuo  $P(t)$

Figura 2.1: Esquema de CHC

### 2.2.2. Búsqueda Dispersa (BD)

La Búsqueda Dispersa (BD) [Glo77, Glo98a, LM03] es un AE que presenta una característica interesante para nuestro estudio que merece la pena destacar: Es el único de los considerados que ha sido diseñado expresamente para poder ser usado de forma combinada con un método de mejora sobre los nuevos individuos generados. Aunque, por definición, no es necesario el uso de un método de mejora, está contemplado en su diseño, y su uso es generalizado. Por lo tanto, aunque por definición la BD no es un AM, en la mayoría de los casos, sí se comporta como tal.

Al igual que en otros AEs (como CHC o los AGs) el principal operador de búsqueda es un mecanismo de combinación de soluciones, equivalente al cruce en los AGs [Glo98b], ya que es el método responsable de utilizar la información disponible de las muestras anteriores para orientar la búsqueda. El procedimiento de mejora puede permitir el uso de herramientas como memorias recientes o memorias intermedias (como la Búsqueda Tabú, [GL97]), el uso de vecindarios variables [HM01], o vecindarios obtenidos mediante métodos de muestreo. En general, se puede utilizar cualquier método heurístico sofisticado de búsqueda [GK03]. El diseño de la BD intenta alcanzar un adecuado equilibrio entre los factores de exploración y explotación en la búsqueda mediante el uso combinado entre el proceso de mejora y el método de combinación.

Aunque inicialmente se presentó la BD como una metaheurística de optimización para problemas de programación entera [Glo77], su enfoque se ha refinado tanto para problemas de optimización discreta como continua [LM03, TK02]. La BD para optimización continua trabaja con vectores de valores reales que combina para producir nuevos individuos mediante generaciones sucesivas, y emplea métodos de BL para optimización continua.

La BD empleada está basada en el esquema indicado gráficamente en la Figura 2.2 y con mayor detalle en la Figura 2.3 [LM03], donde  $P$  determina el conjunto de soluciones generado mediante el método de generación de diversificación;  $RefSet$  es el conjunto de soluciones del conjunto de referencia; y  $Pool$  es el conjunto de soluciones finales obtenidas mediante los métodos de combinación y mejora.  $Psize$  es el tamaño de  $P$  ( $Psize = 100$ ),  $b_1$  es el número inicial de soluciones de alta calidad del conjunto de referencia ( $b_1 = 10$ ),  $b_2$  es el número de soluciones de alta diversidad del conjunto de referencia ( $b_2 = 10$ ), y  $d(\cdot, \cdot)$  es una medida de similitud entre dos soluciones (Hemos empleado la distancia euclídea).

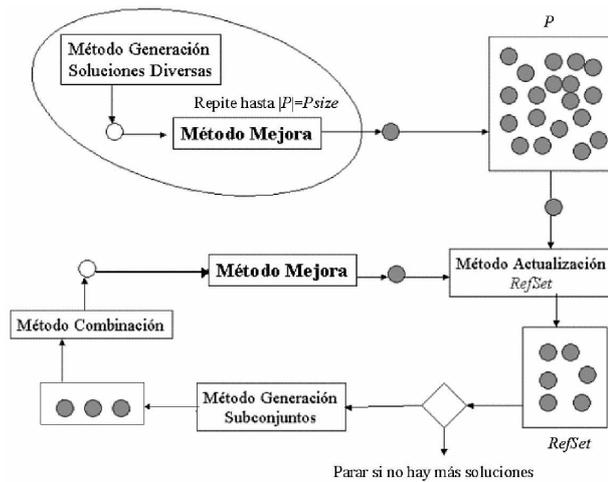


Figura 2.2: Esquema de la Búsqueda Dispersa

El hecho de que se haya diseñado con vista a su integración con un procedimiento independiente de mejora de las soluciones origina determinadas decisiones de diseño que merecen destacarse:

- Tamaño de la población. Frente a los otros algoritmos considerados en los que se recomienda una población mayor (de 50-60 individuos), el tamaño efectivo de la población que evoluciona, el conjunto de referencia *RefSet*, es mucho menor ( $b_1 + b_2 = 20$ ). Esto está en consonancia con la literatura de AMs, en los que se recomienda un menor tamaño de la población [Sin99], con técnicas que puedan mantener diversidad, como las de nichos [CSM04, LW02] o estrategias de mutación o reemplazo inteligentes [EHM99, SS06].
- No define un único conjunto de población a evolucionar, sino que define dos subgrupos con distintos criterios de reemplazo.
  - El primero,  $b_1$ , está formado por las mejores soluciones encontradas hasta el momento, y se actualizan siguiendo un criterio de competencia usando el valor de la función objetivo como criterio (una nueva solución se introducirá en ese conjunto sustituyendo la solución con peor valor si la mejora).

### Algoritmo de Búsqueda Dispersa

1.  $P = \emptyset$ .
2. Repetir hasta que  $|P| = PSize$ 
  - 2.1. Utilizar el **Método de Generación de Soluciones Diversificadas** para obtener una solución  $x$ .
  - 2.2. Aplicar el **Método de Mejora** sobre  $x$  para generar la solución  $x^*$ .
  - 2.3. Si  $x^* \notin P$  entonces  $P = P \cup \{x^*\}$ .
3. Ordenar las soluciones de  $P$  en función de la función objetivo (primero las mejores). Añadir las primeras  $b_1$  soluciones a *RefSet* y borrarlas de  $P$ .
4. Para  $Iter = 1$  hasta *MaxIter* Hacer
  - 4.1. Repetir  $b_2$  veces.
    - Seleccionar la solución  $x' \in P - RefSet$  Que maximice  $d_{min}(x) = \min_{x \in RefSet} d(x, y)$ .
    - $RefSet = RefSet \cup \{x'\}$  y  $P = P - \{x'\}$
  - 4.2. *NewElements* = VERDADERO y *Pool* =  $\emptyset$ .
  - 4.3. Mientras (*NewElements* sea VERDADERO) Hacer
    - Generar *SubSets* mediante el **Método de Generación del Subconjunto**. *NewElements* = FALSO.
    - Mientras (*SubSets*  $\neq \emptyset$ ) Hacer
      - ◊ Seleccionar el siguiente subconjunto (pareja)  $s \in SubSets$ .
      - ◊ Aplicar el **Método de Combinación de Soluciones** a  $s$  para producir una nueva solución,  $x$ , y aplicar el **Método de Mejora** a la solución  $x$  para obtener la solución  $x^*$ .
      - ◊  $Pool = Pool \cup \{x^*\}$  y  $SubSets = SubSets - \{s\}$ .
    - *RefSet* = selección de las  $b$  mejores soluciones de  $RefSet \cup Pool$ .
    - *NewElements* = (*RefSet* tiene al menos una nueva solución).
  - 4.4. Si ( $Iter < MaxIter$ ) Entonces
    - Borrar las últimas (peores)  $b_2$  soluciones de *RefSet*.
    - Construir un nuevo conjunto  $P$  utilizando el **Método de Generación de Soluciones Diversificadas**.

Figura 2.3: Seudocódigo de la Búsqueda Dispersa

Este subconjunto se comporta de forma equivalente a las poblaciones del resto de AE considerados.

- El segundo,  $b_2$ , formado por las soluciones más distantes al primer grupo.

El motivo de este subgrupo está muy influido por la característica anterior. El uso de una pequeña población es recomendable para los AMs, pero el uso de una población reducida de mejores soluciones produciría una escasa diversidad. Dicha falta de diversidad sería un grave inconveniente ya que, no sólo el riesgo de convergencia prematura sería muy alto, sino que muy probablemente las distintas soluciones estarían en la misma base de atracción.

- Aplica un procedimiento sistemático de combinación de soluciones. La BD suele aplicar un método de combinación de soluciones determinístico junto con un método de selección de padres exhaustivo (se cruza toda nueva pareja de soluciones).

En resumen, la BD se suele aplicar como AM, y ha sido empleado con éxito en un gran número de casos [LM03]. Al haber sido diseñada teniendo en cuenta la conveniencia de aplicar la BL, es una propuesta muy interesante a tener en cuenta.

### 2.2.3. Sistemas de Partículas: El modelo *PSO*

Los sistemas de partículas (*Particle Swarm Optimization*, PSO) constituyen una categoría dentro de los AE inspirado en los movimientos naturales de grupos de animales gregarios (bandadas de pájaros, grupos de peces, ...) [KE95, SE98]. En dichos sistemas de movimiento, cada unidad (pájaro) posee un movimiento derivado de dos direcciones: Por un lado, todos los pájaros siguen una misma dirección, siguiendo el del líder de la bandada, y por otro, cada uno posee un movimiento independiente con su propia dirección e inercia.

PSO se caracteriza en que:

- Cada solución es representado por una partícula, que posee una posición (representación de la solución), y una velocidad.

- La exploración es realizada por el movimiento de cada partícula, dirigido por dos componentes: Un vector de inercia (independiente por cada partícula), y un movimiento para acercarla hacia la posición de los mejores individuos: Aquella partícula que haya obtenido el mejor resultado durante todo el proceso de búsqueda; y, para cada solución, la posición anterior en donde haya obtenido un mejor valor.

Matemáticamente, la posición  $X_i(d)$  y la velocidad  $V_i(d)$  que actualizan el valor de la posición  $d$  para el individuo  $i$ , se pueden expresar mediante las Ecuaciones 2.1 y 2.2.

$$V_i(d) = \varpi V_i(d) + c_1 \cdot rand_1() \cdot (Pbest_i(d) - X_i(d)) + c_2 \cdot rand_2() \cdot (Gbest_i(d) - X_i(d)) \quad (2.1)$$

$$X_i(d) = X_i(d) + V_i(d) \quad (2.2)$$

Donde  $\varpi$  es una variable de inercia;  $rand_1$  y  $rand_2$  son dos funciones que devuelven un valor aleatorio entre  $[0,1]$ ;  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  es la posición del  $i$ -ésimo individuo;  $Pbest_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{iD})$  es la mejor solución encontrada por la partícula  $i$ ; y  $Gbest_i = (gbest_{i1}, gbest_{i2}, \dots, gbest_{iD})$  es la mejor solución actual encontrada hasta el momento durante la búsqueda.

El considerar la mejor posición encontrada hasta el momento permite dirigir el conjunto de soluciones hacia la región considerada como más prometedora (exploración), y el orientar cada partícula hacia la mejor posición encontrada por ella misma, permite realizar una búsqueda cercana a dicha posición (explotación de las mejores soluciones encontradas).

Estos algoritmos permiten una alta velocidad de convergencia, presentando tendencia a una convergencia prematura. Esta tendencia se vería aumentada por la incorporación de un método de BL, por lo que es necesario incorporar sobre el modelo original variaciones que aporten diversidad. Un medio para aumentar la diversidad es por medio del uso del concepto de vecindario [Ken99], que permite guiarse en vez de por la mejor de todas las soluciones encontradas, por la mejor encontrada por un subconjunto de partículas. Es decir, usan un  $Gbest$  distinto para cada partícula, definido como como el mejor de las  $Pbest$  encontradas por un subconjunto de partículas (distinto para cada una). El uso de vecindarios permite aumentar la diversidad. Existen modelos meméticos de  $PSOs$  que usan estos modelos [LS05].

En nuestro caso, utilizaremos como *PSO* candidato el CLPSO [LQSB03]. En este algoritmo la Ecuación 2.1 es reemplazada por 2.3.

$$V_i(d) = \varpi V_i(d) + c_1 \cdot \text{rand}_1() \cdot (Pbest_i(d) - X_i(d)) + c_2 \cdot \text{rand}_2() \cdot (Gbest'_i(d) - X_i(d)) \quad (2.3)$$

En donde  $Gbest'_i(d) = Pbest_j(d)$  donde  $j \in [1, Popsize]$ , elegido aleatoriamente de entre el total de soluciones. El valor  $Gbest'_i$  elegido no es el mismo para cada dimensión, sino que es elegido aleatoriamente por cada una.

Se puede observar que  $Gbest$  (es decir, la mejor solución encontrada hasta ahora) no se usa explícitamente, pero dado que  $Gbest$  es el  $Pbest$  de alguna partícula, tiene las mismas posibilidades de ser utilizadas en la búsqueda que el resto.

Usaremos el CLPSO ya que, no sólo ofrece mejores resultados que los PSOs más clásicos [LQSB03], sino que mediante este comportamiento se reduce el riesgo de convergencia prematura.

Durante los experimentos seguiremos el modelo de hibridación clásico (el de Hart) porque es necesario para el análisis comparativo, como MeSwarm [LCC<sup>+</sup>05]. Sin embargo, este es un modelo que presenta problemas con el PSO:

- Por un lado, es  $V_i$  el que determina la siguiente solución. Si la mejoramos posteriormente mediante un proceso de BL sin alterar dicho valor, se podría mantener por inercia una dirección de búsqueda contraria a la indicada por la BL. Para evitarlo, cada vez que se aplicó la BL, tras aplicar la BL se alteró  $V_i$  mediante la Ecuación 2.4.

$$V_i(d) = V_i(d) + (X_{new}(d) - X_{old}(d)) \quad (2.4)$$

Donde  $X_{old}$  y  $X_{new}$  son los valores de la partícula antes y después de aplicar la BL, respectivamente.

De esta forma, se consigue que se mantenga la correspondencia entre la posición actual de una partícula y su velocidad.

- Por el otro, el uso del  $Pbest$  permite realizar ya una cierta explotación de las mejores soluciones encontradas, por lo que la BL puede no ofrecer el grado de mejora deseado.

Esta serie de inconvenientes hace que algunos AMs que hacen uso del PSO (como LianPSO [LS05]) sigan enfoques distintos (como aplicar la BL sólo hacia el mejor, para que éste guíe al resto de partículas). Sin embargo, en nuestro caso, para poder comparar los AEs entre sí, necesitamos aplicar el mismo criterio de hibridación.

#### 2.2.4. Algoritmos Genéticos: Modelo Generacional y Estacionario

Existen dos esquema muy distintos que estudiaremos de forma separada:

- Modelo Generacional.

En el modelo generacional se sigue un proceso de reemplazo generacional, en el que se reemplaza la población entera por una nueva en cada generación. La nueva población generada está formada por todos los descendientes generados y por un conjunto de los antecesores (para mantener constante el tamaño de la población). Por tanto, todo nuevo individuo generado es almacenado en la siguiente población, independientemente de su bondad.

La Figura 2.4 muestra el esquema general del AGG.

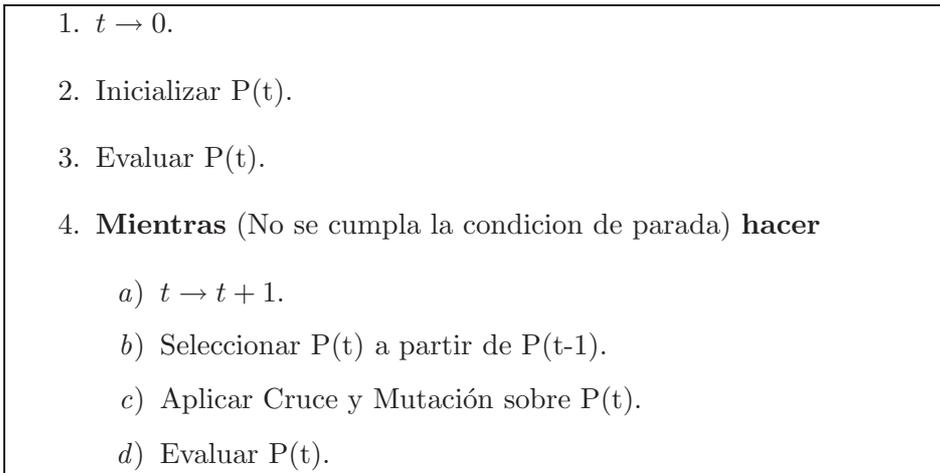


Figura 2.4: Estructura de un AGG

Dado que la nueva población reemplaza a la anterior, las soluciones mejora-

das por la BL pueden no perdurar en la población, por lo que esas regiones explotadas por la BL podrían no volver a ser exploradas por el AGG, alternándose fases de búsqueda global y local. Primero, el AGG produce una nueva población, y luego, la BL es realizada sobre un subconjunto de estos individuos. Aunque el resultado de la BL puede no mantenerse entre poblaciones, puede influir en la selección de padres para los cruces de la siguiente generación.

El hecho de que soluciones con alto valor en la función objetivo puedan ser reemplazadas por soluciones con peor valor, supone un lastre para su uso en AMs, al no garantizarse que se mantenga la mejora obtenida por la BL, tal y como se recomienda en este tipo de algoritmos [Har94].

■ Modelo Estacionario.

En el modelo estacionario se produce únicamente uno o dos hijos en cada generación. En primer lugar, se seleccionan los individuos que se cruzarán para producir nuevos individuos. Después, se aplica una estrategia para decidir cuáles van a ser sustituidos por los nuevos cromosomas. La figura 2.2.4 muestra los pasos básicos del AG estacionario (AGE). Estos pasos son repetidos hasta que se alcanza una condición de terminación (en nuestro caso, alcanzar un número máximo de evaluaciones).

1. Seleccionar una pareja de padres de la población.
2. Crear un descendiente mediante cruce y mutación.
3. Evaluarlo con la función objetivo, y en su caso, aplicar la BL.
4. Seleccionar el individuo de la población que pueden reemplazarse por los descendientes.
5. Decidir si el individuo finalmente será reemplazado.

Figura 2.5: Estructura del ciclo evolutivo de AGE

En el paso 1 se determina la elección de los padres, que permite ajustar el equilibrio entre explotación (favoreciendo el cruce entre soluciones con alto valor de la función objetivo) y exploración (eligiendo soluciones distintas entre sí). En el paso 3, se puede, opcionalmente, aplicar la BL para mejorar

individuos obtenidos en el paso anterior. En el paso 4, se puede elegir la *estrategia de reemplazo* (descritas en el apartado 2.2.5.5), y en el paso 5 se determina el criterio de reemplazo.

Aunque los AGEs son menos utilizados que los AGGs, Land [Lan98] recomendó su uso para la construcción de *AMs estacionarios* (AGEs + BL) porque pueden ser más estables (ya que las mejores soluciones no son reemplazadas hasta que las nuevas generadas son mejores), y permiten que los resultados de la BL sean mantenidos en la población, influyendo en las generaciones posteriores. Esto permitiría que los componentes de exploración del AGE y la explotación de la BL se influyan mutuamente.

Es un modelo interesante desde un punto de vista de un AM, por dos motivos. Por un lado, por su flexibilidad, derivada de la posibilidad de elegir de entre diferentes estrategias de selección de padres y de reemplazo; por el otro, por el uso de solapamiento en la población (que establece la competencia como criterio de reemplazo), que permite que un individuo con un buen valor para la función objetivo perdure durante un cierto tiempo en la población.

### 2.2.5. Operadores Comunes

En este apartado vamos a comentar los distintos componentes comunes a los distintos AEs. Estos operadores poseen una gran influencia, por lo que a la hora de analizar los distintos algoritmos, realizaremos las comparaciones correspondientes para cada uno de estos operadores.

#### 2.2.5.1. Operadores de cruce

El operador de cruce es uno de los operadores más importantes, ya que es el principal operador que introduce diversidad en la mayoría de los AEs considerados (CHC, AG, BD). El cruce, también llamado método de combinación de soluciones, es un método que permite compartir información entre soluciones. Se basa en combinar características de los vectores de varios padres (usualmente dos) para producir nuevas soluciones (denominada *descendencia*). El operador de cruce parte de la heurística de que de la combinación de buenas soluciones se pueden producir soluciones mejores.

A la hora de realizar el estudio comparativo, consideraremos dos operadores

de cruce distintos: el operador  $BLX - \alpha$ , considerado como uno de los operadores de cruce más efectivo para el AGCR [HLS03]; y el operador de cruce aritmético (muy empleado para ciertos AEs como la BD).

Sean  $C_1 = (c_1^1, \dots, c_n^1)$  y  $C_2 = (c_1^2, \dots, c_n^2)$  dos cromosomas que han sido seleccionados para aplicar sobre ellos el operador de cruce correspondiente, a continuación se describe la forma en la que cada operador de cruce trabaja.

- El operador de cruce  $BLX - \alpha$  [BRS66, ECS93]. Se genera un descendiente  $H = (h_1, \dots, h_n)$ , en donde  $h_i$  es generado de manera (uniformemente) aleatoria dentro del intervalo  $[c_{min} - I \cdot \alpha, c_{max} + I \cdot \alpha]$ , donde  $c_{max} = \max(c_i^1, c_i^2)$ ,  $c_{min} = \min(c_i^1, c_i^2)$ , y  $I = c_{max} - c_{min}$ . Se muestra visualmente en la Figura 2.6.

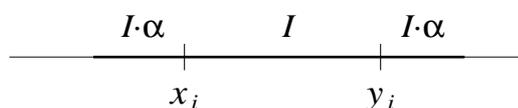


Figura 2.6: Método de combinación  $BLX - \alpha$

- El operador de cruce aritmético [Mic96]. En éste, como en todo operador geométrico, se calcula el descendiente generado  $H = (h_1, \dots, h_n)$  de forma determinística (no hay componente aleatorio) mediante un operador geométrico. En el caso del operador aritmético, dicho operador es la media geométrica, por lo que  $h_i = (c_i^1 + c_i^2)/2$ , tal y como se muestra en la Figura 2.7.

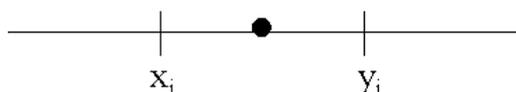


Figura 2.7: Método de combinación Aritmético

A continuación comentamos algunas de las principales características y diferencias entre ambos operadores de cruce:

- Entre los operadores de cruce existe una diferencia muy importante: El operador  $BLX - \alpha$ , a diferencia del operador aritmético, es un operador con un componente aleatorio. Esto quiere decir que, dado un mismo par de padres el operador de cruce puede generar un conjunto de soluciones distintas, mientras que en el operador aritmético, dado el mismo par de soluciones padres, siempre se genera el mismo descendiente.

Para ciertos algoritmos, como la BD, tradicionalmente se han aplicado operadores de cruce no aleatorios o deterministas, aunque se ha mostrado que el uso de operadores de combinación que introducen aleatoriedad puede ser muy efectivo [LM03],[LHKM04].

- En el operador de cruce aritmético siempre se reduce el radio de distribución de las nuevas soluciones, mientras que, en el operador de cruce  $BLX - \alpha$ , el parámetro  $\alpha$  determina el grado de distribución de los nuevos individuos, pudiéndose reducir o aumentar la diversidad de la población. Es más, el operador de cruce aritmético puede verse como un caso particular de  $BLX - \alpha$ , en el que  $\alpha = 0$ .

En el trabajo de Nomura y Shimohara [NS01] se demostró que  $BLX - \alpha$  posee la habilidad de promover diversidad en la población del AE. Más específicamente, estos autores presentaron un modelo teórico bajo el que analizaron la relación existente entre las funciones probabilísticas de densidad de población, antes y después de su aplicación, bajo la hipótesis de una población de longitud infinita. Su conclusión, verificada posteriormente mediante simulación, fue que el operador  $BLX - \alpha$  aumenta la diversidad cuando  $\alpha > \frac{\sqrt{3}-1}{2}$ , reduciéndose en cualquier otro caso (con valores de  $\alpha$  menores). Posteriormente, Beyer y Deb [BD01] concluyeron que para una población de  $\mu$  individuos, si  $\alpha > \frac{\sqrt{3-\mu/6}-1}{2}$  el cruce aumenta la diversidad de la población.

De ambos trabajos se desprende que el uso de  $BLX - \alpha$  con valor de  $\alpha=0.3$  sería un valor para mantener (relativamente) la diversidad, mientras que un valor  $\alpha=0.5$  produciría diversidad (aumentando la distribución de hijos respecto a los padres). Ambos valores van a ser utilizados en nuestros experimentos.

- $BLX - \alpha$  tiene un comportamiento *autoadaptativo* en el que la generación de los descendientes se realiza en función de la distribución de los padres, adaptándose a ésta, sin necesidad de ningún parámetro adaptativo. El parámetro aritmético adolece de dicho comportamiento.

Beyer y Deb [BD01] afirman que es esencial que los operadores de variación (que introducen nuevas soluciones o modifican las existentes) consideren la distribución de los padres en el espacio de búsqueda, ya que, en ese caso, el AE resultante ofrece un comportamiento autoadaptativo.  $BLX - \alpha$  utiliza distribuciones de probabilidad que se calculan considerando la distancia entre las variables de decisión de los padres ( $C^1$  y  $C^2$ ). Si los padres están próximos entre sí, los nuevos individuos generados estarán distribuido densamente alrededor de éstos. Si, por el contrario, están alejados entre sí, sus hijos estarán distribuidos alrededor de éstos de forma dispersa. Este comportamiento implica que el operador de cruce  $BLX - \alpha$  permite al AE converger, diverger o adaptarse a regiones con distinto resultado de la función objetivo sin que sea necesario incorporar nuevos métodos o parámetros para alcanzar dicho comportamiento. Es más, en el pasado reciente, los AGCRs con  $BLX - \alpha$  han demostrado un comportamiento autoadaptativo similar al observado en otros enfoques, como la estrategias de evolución o la programación evolutiva [BD01].

### 2.2.5.2. Operadores de mutación

El operador de mutación es un operador capaz de introducir de forma aleatoria diversidad en la población a partir de una única solución. Su uso es muy variado, aunque en nuestro caso sólo los AGs emplean un operador de mutación como método para fomentar la diversidad de la búsqueda. El operador de mutación es un operador que modifica uno o varios genes de una solución de la población, dicha modificación se utiliza para aumentar la diversidad de las poblaciones.

A la hora de realizar el análisis consideraremos dos operadores de mutación distintos, el operador de mutación no uniforme, y el operador de mutación BGA [MSV93], que es un operador de mutación mucho más fuerte. A continuación definimos los distintos tipos de mutación:

Sea  $[a, b]$  el intervalo de búsqueda, se selecciona aleatoriamente el gen  $i$ -ésimo de la solución  $C = (c_1, \dots, c_n)$  para ser mutado, obteniéndose el gen  $c'_i$  según la expresión:

- *Mutación no uniforme* [Mic92]:

$$c'_i = \begin{cases} c_i + (b_i - c_i) \cdot \Gamma(t), & \text{si } random_1 < 0,5 \\ c_i - (c_i - a_i) \cdot \Gamma(t), & \text{si } random_1 \leq 0,5 \end{cases} \quad (2.5)$$

En donde

$$\Gamma(t) = (random_2 (1 - t/t_{max}))^d$$

$random_1$  y  $random_2$  son dos números aleatorios generados en el intervalo  $[0, 1]$ ,  $d$  es un parámetro de refinación,  $t$  es el número de generación (o evaluación) y  $t_{max}$  el máximo número de generaciones (o evaluaciones) permitido.

- *Mutación BGA* [MSV93]:

$$c'_i = c_i \pm rang_i \cdot \sum_{k=0}^{15} \alpha_k 2^{-k} \quad (2.6)$$

En donde  $rang_i$  define el rango de mutación, se ha aplicado el valor usual  $0,1 \cdot (b_i - a_i)$ . El signo (+ ó -) es elegido cada vez con una probabilidad de 0,5 y  $\alpha_k$  toma para cada variable un valor 0 ó 1 determinado aleatoriamente, mediante la expresión  $p(\alpha_i = 1) = \frac{1}{16}$ .

Existen importantes diferencias entre ambos tipos de mutación:

- Tratan de forma distinta el *tamaño de salto*, que posee gran influencia [HL00]: Mientras que el operador no uniforme lo adapta (mayor al principio y menor conforme el algoritmo avanza), en el operador BGA dicho factor se mantiene constante.

El hecho de adaptar dicha distancia es una estrategia empleada para fomentar la diversidad en las etapas iniciales, y pasar posteriormente a centrarse en un mayor ajuste local. Este enfoque se ha mostrado como uno de los enfoques más adecuados para AGCR [HLV98]. Sin embargo, cuando se utiliza el AE combinado con un método de BL, esta necesidad de un ajuste cada vez más fino es realizado por dicho componente, por lo que un operador de mutación que mantenga constante el tamaño del cambio durante toda la búsqueda, como hace el operador de mutación BGA, podría suponer una ventaja.

- Mientras que en el operador no uniforme el gen generado puede tomar valores dentro de todo el rango  $[a_i, b_i]$ , en el operador BGA toma valores dentro del intervalo  $[c_i - rang_i, c_i + rang_i]$ , con lo que la probabilidad de generar un vecino de  $c_i$  es muy alta (la proximidad mínima es del orden de  $rang_i \cdot 2^{-15}$ ).

### 2.2.5.3. Métodos de Búsqueda Local Empleados

En este apartado incluimos una descripción detallada de los dos métodos de BL empleados en el estudio comparativo. Estos métodos de BL son clásicos y bien conocidos en la bibliografía: El algoritmo Solis y Wets (SW) [SW81] y el método Simplex de Nelder y Mead (S) [NM65]. Ambos son métodos aleatorios con estructura de ascensión de colinas: siguen un proceso iterativo, en el que en cada paso se genera una nueva solución a partir de una solución actual, que reemplazará a la anterior en el caso de que lo mejore. El proceso anterior se repite hasta que se alcanza un criterio de parada, que en nuestro caso es un número máximo de evaluaciones de la función objetivo durante la BL (denominada intensidad de la BL,  $n_{int}$ ). Esta estructura les permite buscar eficientemente un óptimo local cercana a una solución ya dada.

Se caracterizan ambos en que no requieren conocer más información que los propios valores de la función (como el valor de la derivada en un punto) como es necesario con otros métodos de BL, por lo que su capacidad de aplicación es muy alta.

A continuación, presentamos una descripción detallada de estos procedimientos.

**Método Solis-Wets.** Este método de BL aleatorio sigue un esquema de ascensión de colinas con un tamaño de salto adaptativo. En cada iteración se parte de una solución actual  $x$ . Se genera un valor  $d$  mediante una distribución normal con 0 de media y  $\rho$  de desviación estándar. La nueva solución se obtiene sumando dicho valor  $d$  junto con un valor *bias* que mantiene un cierto grado de 'inercia' sobre los movimientos anteriores.

Se caracteriza en varios aspectos:

- Las nuevas soluciones se obtienen mediante la suma de una variable de incremento (generada mediante una distribución normal  $N(0, \rho)$ ) y una

variable *bias*. Esta variable *bias* permite mantener una inercia (orientando la búsqueda hacia direcciones exitosas en iteraciones anteriores).

- En cada paso considera una dirección y si el resultado no lo mejora, la contraria.
- Cuenta el número de éxitos (soluciones que mejoran a la actual) o fallos (soluciones peores que la actual) consecutivos. En función de unos valores máximos, el parámetro  $\rho$  es incrementado o decrementado, para aumentar o disminuir la búsqueda sobre la solución actual.

En la Figura 2.8 aparece el esquema completo. Puede consultar [SW81] para una descripción más detallada.

**Método Simplex Nelder-Mead.** Este, al igual que en el anterior, es un método clásico con un esquema descendente. Se denomina Simplex a una figura geométrica en  $n$  dimensiones compuesta de  $n + 1$  puntos  $s_0 \cdots s_n$ . Uno de los puntos del Simplex es la solución actual, y los otros  $n$  puntos se utilizan para definir un vector de direcciones a lo largo de las  $n$  dimensiones. Es decir, sea  $s_0$  el punto inicial, los  $n$  restantes puntos  $s_i$  son generados mediante la expresión  $s_i = s_0 + \lambda e_j$ , donde los  $e_j$  son  $n$  vectores unitarios (uno en cada dimensión), y  $\lambda$  es una constante que es generalmente igual a la unidad (aunque podría ser adaptada a las características específicas del problema).

Sobre dicha estructura simplex se define una serie de transformaciones geométricas elementales (reflexión, contracción, expansión y multi-contracción), que permitan mover, expandir o contraer el simplex. En cada paso, para seleccionar la operación a aplicar, el método utiliza los valores de la función a optimizar para los vértices del simplex considerado. Después de aplicar cada transformación el considerado peor vértice será sustituido por un nuevo vértice con mejor valor para la función objetivo.

A continuación procedemos a explicar por encima el proceso, en donde la función  $Fitness(x)$  representa el valor de la función objetivo de la solución  $x$ ,  $w$  representa el vértice de simplex con peor valor de la función objetivo, y  $v$  a al mejor vértice.

Al principio del algoritmo  $w$  es reemplazado por una nueva solución  $x$ , denominada imagen, generada mediante el operador de reflexión. Si el punto reflejado es mejor que el resto de puntos, se aplica el método de expansión sobre esa dirección. Sino es así:

### Método Solis Wets

1.  $fitness\_actual \leftarrow evaluacion(x)$ .  $n_{eval} \leftarrow 1$ .
2. Inicializa  $\rho$ .
3.  $bias \leftarrow 0$ .
4.  $numExitos \leftarrow 0$ .  $numFallos \leftarrow 0$ .
5. Mientras  $n_{eval} < n_{int}$ 
  - 5.1.  $d \sim N(0, \rho)$ .
  - 5.2.  $x' \leftarrow x + bias + d$ .
  - 5.3.  $fitness\_x' \leftarrow evaluacion(x')$ .  $n_{eval} \leftarrow n_{eval} + 1$ .
  - 5.4. Si  $fitness\_x' < fitness\_actual$ 
    - 5.4.1.  $x \leftarrow x'$ .
    - 5.4.2.  $bias \leftarrow 0,2 \cdot bias + 0,4 \cdot (d + bias)$ .
    - 5.4.3.  $numExitos \leftarrow numExitos + 1$ ;  $numFallos \leftarrow 0$ .
  - 5.5. En otro caso
    - 5.5.1.  $x'' \leftarrow x - bias - d$ .
    - 5.5.2.  $fitness\_x'' \leftarrow evaluacion(x'')$ .  $n_{eval} \leftarrow n_{eval} + 1$ .
    - 5.5.3. Si  $fitness\_x'' < fitness\_actual$ 
      - ◊  $x \leftarrow x''$ .
      - ◊  $bias \leftarrow bias - 0,4 \cdot (d + bias)$ .
      - ◊  $numExitos \leftarrow numExitos + 1$ ;  $numFallos \leftarrow 0$ .
    - 5.5.4. Si no
      - ◊  $bias \leftarrow 0,5 \cdot bias$ .
      - ◊  $numExitos \leftarrow 0$ ;  $numFallos \leftarrow numFallos + 1$
  - 5.6. Si  $numExitos \geq maxExitos$  ( $maxExitos = 5$ )
    - $numExitos \leftarrow 0$ .
    - $\rho \leftarrow 2 \cdot \rho$ .
  - 5.7. Si  $numFallos \geq maxFallos$  ( $maxFallos = 3$ )
    - $numFallos \leftarrow 0$ .
    - $\rho \leftarrow 0,5 \cdot \rho$ .

Figura 2.8: Esquema del método de BL Solis Wets

1. Se aplica el operador de reflexión sobre el vértice con peor valor de la función objetivo  $w$ , obteniendo una nueva solución imagen  $N$ .
2. Mientras no se termine el criterio de parada:
  - Si  $Fitness(N) \leq Fitness(v_i) \forall i \in [0, \dots, n]$ , se aplica el método de expansión sobre la dirección de  $N$ .
  - En caso contrario, y Si  $Fitness(N) \leq Fitness(w)$ , se sustituye  $w$  por  $N$ , y se aplica de nuevo el operador de reflexión, esta vez sobre el nuevo peor vértice.
  - Si  $Fitness(N) \geq Fitness(w)$  se aplica el operador de contracción. La idea es que el simplex se *autoadapte* a la superficie de la función. Si el nuevo valor obtenido por contracción sigue siendo peor que  $w$ , se aplica una multi-contracción.
3. Devolver el vértice con mejor valor para la función objetivo.

La Figura 2.9 muestra como ejemplo las distintas operaciones sobre un simplex tridimensional. En la parte superior se muestra el simplex identificando al peor vértice. Los distintos apartados (del *a* a *d*) muestran visualmente el estado si se realizase cada una de las operaciones de reflexión, expansión, contracción y múltiple contracción. Esta figura ha sido obtenida del conocido libro *Numerical Recipe* [PTVF92]. Puede consultar [NM65] para una descripción más detallada.

#### 2.2.5.4. Métodos de Selección de Padres

Los métodos de selección de padres son básicos dentro de los AGs, ya que determinan las posibilidades de los individuos de influir en las generaciones siguientes. Dado que en los AGs el operador que produce diversidad es el cruce, una adecuada selección de los individuos a cruzar es fundamental para poder conseguir un adecuado equilibrio entre exploración y explotación.

A continuación se comentan los métodos de selección de padres utilizados en nuestros análisis.

- Selección por Torneo (*Tournament Selection*, TS). Se muestrea aleatoriamente un grupo de  $N_{TS}$  individuos de la población y se selecciona el que posea el mejor valor para la función objetivo.

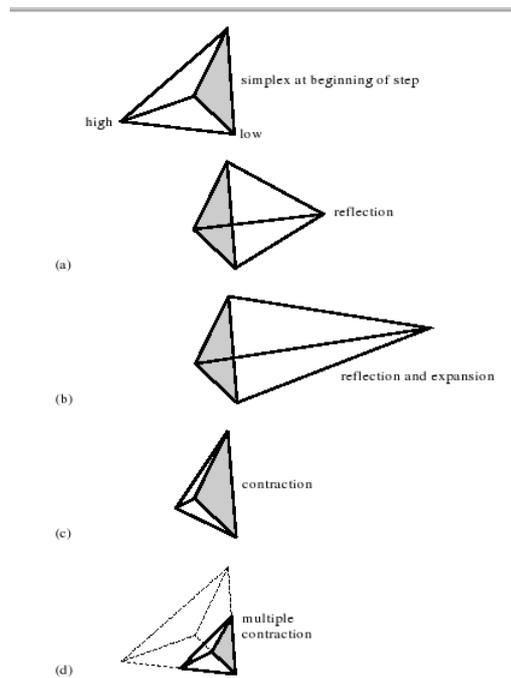


Figura 2.9: Operadores de Transformación para el método *Simplex*

El valor  $N_{TS}$  regula el nivel de presión selectiva, cuanto mayor sea  $N_{TS}$  mayor número de individuos compiten entre sí y por tanto el valor de la función objetivo cobra mayor importancia, aumentando la presión selectiva. Por contra, cuanto menor sea  $N_{TS}$  más fácil será para un individuo con escasa bondad ser seleccionado (al competir sólo con un número pequeño de individuos). Los casos extremos serían que  $N_{TS}$  fuese igual al tamaño de la población (en cuyo caso el o los dos mejores individuos serían siempre seleccionados), o que  $N_{TS}$  fuese igual a uno (totalmente aleatorio).

La selección por torneo es un método clásico que presenta un problema importante: El problema de la escala. Este problema consiste en que, al competir los individuos por su valor para la función objetivo, el número de individuos con un valor alto son elegidos con mucha frecuencia, mientras que el resto son seleccionados muy raramente. La consecuencia de este problema es una falta de diversidad en la exploración, ya que grupos numerosos de soluciones (que pueden representar a regiones distantes) no son utilizadas

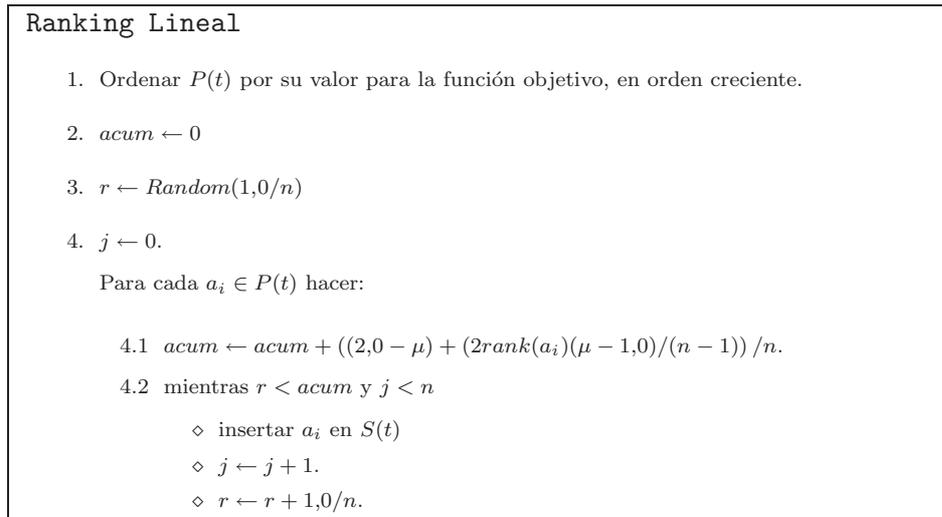


Figura 2.10: Esquema del Ranking Lineal

para generar las nuevas soluciones. Para evitarlo se suele aplicar un valor bajo de  $N_{TS}$ . En nuestros experimentos utilizamos  $N_{TS} = 3$ .

- Ranking Lineal (*Lineal Ranking*, LR) [Bak85]. Se ordenan los individuos de la población por su valor objetivo (1 al peor, 2 al segundo peor, ...) y asigna a cada uno un valor de probabilidad de ser elegido como padre. Dicho valor de probabilidad se determina en función del *ranking* asignado (será mayor cuanto mejor sea su posición) multiplicado por un valor de escala  $\mu_{min}$ . La figura 2.10 muestra el esquema, en donde  $P(t)$  es la población actual y  $S(t)$  el conjunto de individuos seleccionados para ser padre.

El uso de los *ranking* permite evitar el problema de escala que puede conducir la selección por torneo. El rango de reproducción se limita para que ningún individuo genere un número excesivo de descendientes. El uso de *ranking* introduce una escala uniforme entre la población, y ofrece un mecanismo simple y efectivo para controlar la presión selectiva. En nuestros experimentos utilizaremos  $\mu_{min}=0.75$ .

- Selección Aleatoria (*Random Selection*, RS). RS selecciona un individuo de la población de forma totalmente aleatoria. Utilizamos este método de selección para poder usarlo como referencia.

- Emparejamiento Variado Inverso (*Negative Assortative Mating*, NAM) [FR01]. Está orientado a generar diversidad. El NAM forma parte de los métodos de emparejamiento variado [FR01]. En ellos, se sigue un método distinto para seleccionar cada uno de los padres. Mediante una ruleta se selecciona el primer padre y  $N_{NAM}$  individuos (aunque en nuestros experimentos todos los individuos son elegidos de forma totalmente aleatoria). Luego, se calcula una medida de similitud entre todos los cromosomas y el primer padre elegido (la similaridad entre dos cromosomas con codificación real es definida como la distancia euclídea entre ellas). A partir de aquí existen dos variantes: El emparejamiento variado directo, que elige al individuo más cercano al primero, o el NAM que elige al individuo más diferente al primero. Claramente, el mecanismo del NAM incrementa la diversidad en la población emparejando con alta probabilidad individuos distintos entre sí. En nuestros experimentos utilizaremos  $N_{NAM} = 3$ .

### 2.2.5.5. Estrategias de Reemplazo

En los AGEs se debe de determinar los individuos que serán reemplazados por los nuevos individuos generados. Dicha elección puede determinar el nivel de presión selectiva que se exige a los individuos para ser introducidos o permanecer en la población, y la pérdida de diversidad que se introduce conforme el proceso de búsqueda avanza. A lo largo del tiempo se han propuesto distintas estrategias de reemplazo, algunas con el objetivo de introducir alta presión selectiva (como el reemplazar el peor de la población), otras para mantener una alta diversidad (como los *métodos de multitud o crowding* que buscan que las nuevas soluciones reemplacen las soluciones parecidas a ellas) o un equilibrio entre ambas.

A continuación se describen las estrategias de reemplazo consideradas. No se ha planteado como un conjunto exhaustivo, sino como un conjunto pequeño pero representativo.

- Reemplazo del Peor (*Replacement Worst*, RW) [GD91]. En éste, se reemplaza el peor elemento de la población si el nuevo hijo lo mejora. Ofrece alta presión selectiva, incluso cuando sus padres son elegidos aleatoriamente. Puede observarse que es el criterio que facilita más la incorporación de un nuevo individuo manteniendo al mismo tiempo un criterio elitista.
- Selección por Torneo (*Restricted Tournament Selection*, RTS) [Har95]. En éste, a la hora de elegir el individuo sustituido por el nuevo individuo A',

muestra aleatoriamente  $w$  individuos de la población y, de entre éstos, selecciona el individuo más cercano a  $A'$  (usando la distancia euclídea como medida).  $A'$  es comparada con el individuo seleccionado y si  $A'$  posee mejor valor para la función objetivo, se le permite entrar en la población, sustituyéndolo. En nuestros experimentos utilizamos un tamaño de ventana  $w = 3$ .

- *Crowding* Determinístico (*Deterministic Crowding*, DC) [Mah92]. Este considera que los padres deben de ser los miembros de la población a ser sustituidos por sus padres. En DC, para el par de padres  $(P_i, P_j)$  que genera los descendientes  $(O_i, O_j)$  compiten éstos con sus padres para ser incluidos en la población según el esquema indicado en la figura 2.11. En nuestro caso, sin embargo, para facilitar la comparación con el resto de técnicas, únicamente un individuo  $O$  es generado en cada paso, que sustituirá a su padre más parecido (según la medida de distancia euclídea) únicamente si posee mejor valor para la función objetivo.

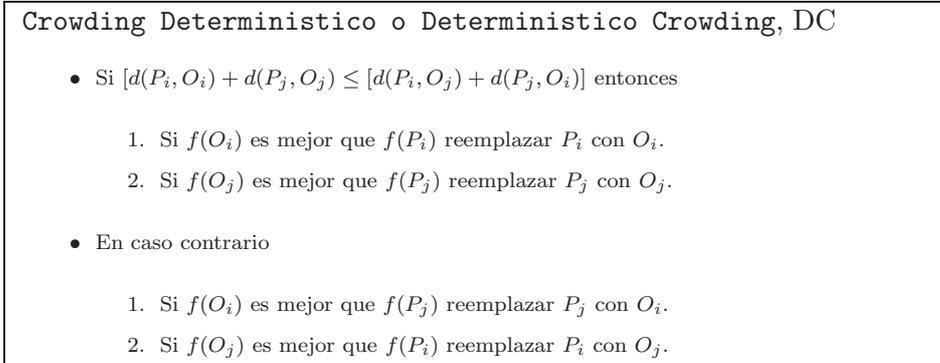


Figura 2.11: Esquema del DC

- Peor entre Semejantes (*Worst Among Most Similar*, WAMS). Esta estrategia se compone de dos pasos. En el primero,  $f$  grupos son creados eligiendo aleatoriamente  $g$  individuos de la población. Luego, para cada grupo se selecciona aquel con mayor similaridad con el nuevo individuo (distancia euclídea). Esto produce  $f$  individuos que son candidatos para el reemplazo, de entre los cuales se seleccionará aquel con peor valor, para ser reemplazado por el nuevo. Esta estrategia está mostrada en la Figura 2.12.

Es la única estrategia de reemplazo de las consideradas que no valora los

valores de la función objetivo, con lo que se permite que soluciones con peor valor para la función objetivo sustituyan a soluciones con mejor valor.

El comportamiento de este algoritmo depende en gran medida de los valores  $f$  y  $g$  [CV99]. En nuestros experimentos usaremos  $f = 6$  y  $g = 9$ .

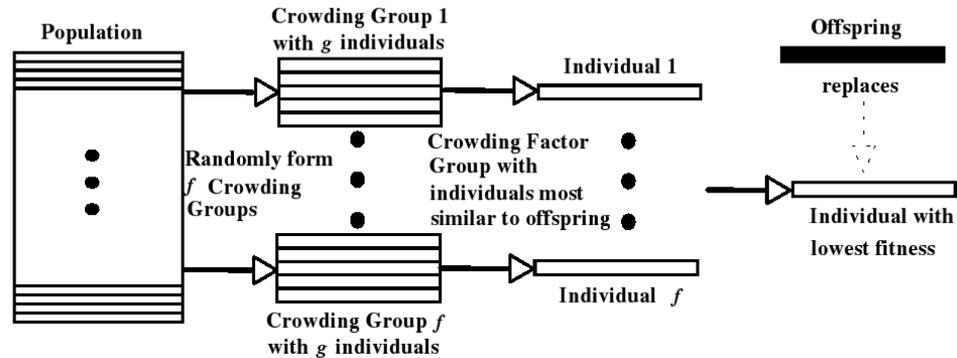


Figura 2.12: Esquema de la estrategia de reemplazo WAMS

### 2.3. Estudio Experimental

En este apartado presentamos los experimentos realizados con objeto de estudiar el comportamiento y conveniencia de cada uno de los distintos AEs considerados.

Esta sección está estructurada de la siguiente forma:

- En la Sección 2.3.1, se describen las principales características del marco experimental usado: funciones empleadas, las características de las ejecuciones, características de los algoritmos empleados en las comparaciones, y los criterios empleados (tests estadísticos o de otro tipo) para identificar los más adecuados.
- En la Sección 2.3.2, se estudia el comportamiento de cada uno de los distintos AEs considerados. Para cada AE se estudia las distintas variantes con objeto de estudiar para cada uno la influencia de la BL (cruce y operadores de selección de padres y estrategias de reemplazo, en los casos deseados):

- En la Sección 2.3.2.1, se compara CHC sobre distintos cruces, para distintas intensidades de BL, y se identifican las mejores combinaciones encontradas.
  - En la Sección 2.3.2.2, se compara BD sobre distintos cruces, para distintas intensidades de BL, y se identifican las mejores combinaciones encontradas.
  - En la Sección 2.3.2.3, se compara CLPSO para distintas intensidades de BL, y se identifican las mejores combinaciones encontradas.
  - En la Sección 2.3.2.4, se compara el AGG para distintas intensidades de BL: Primero, se determina el operador de mutación más adecuado. Luego, se identifican las mejores combinaciones encontradas de cruce y el método de selección.
  - En la Sección 2.3.2.5, se compara el AGE sobre distintas intensidades de BL: Primero, se identifica el operador de cruce más adecuado. Luego, se determinan las mejores combinaciones encontradas de mecanismo de selección de padre y estrategia de reemplazo.
- En la Sección 2.3.3 realizamos un análisis comparativo de los mejores modelos de AEs considerados en la sección anterior.
  - Finalmente, en la Sección 2.4 se presentan las conclusiones final del estudio comparativo.

### 2.3.1. Marco Experimental

En este apartado se describe la información relativa a las condiciones de los experimentos realizados:

- Primero, se describen las distintas funciones de evaluación empleadas en el estudio comparativo.
- Se describen las características (parámetros) de los distintos algoritmos comparados, y las características de los experimentos realizados.
- Se describen los test estadísticos empleados.
- Finalmente, se describen otra medida, no estadística, empleada en nuestro estudio.

**Casos de Tests** El conjunto de casos de tests empleado está compuesto de los siguientes casos:

- 10 funciones de tests conocidas, descritas en la Tabla 2.1. En la tabla 2.2 se especifica para cada una de las funciones de test anteriores el rango de búsqueda empleado y la dimensionalidad.

El óptimo para todas ellas es el vector  $\vec{0}$ , pero para evitar favorecer técnicas de búsqueda que favorezcan la exploración de las zonas centrales, en la mayoría de las funciones dicha solución no se encuentra en el centro del dominio de búsqueda, como se observa en la Tabla 2.2.

La dificultad de las 10 funciones es variable: Desde funciones muy sencillas y unimodales (como la esfera) hasta funciones multimodales (como la Rosenbrock).

- 3 problemas reales: Sistema de Ecuaciones Lineales (SLE), el Problema de Identificación de parámetros sonoros de modulación de frecuencia (FMS) y el problema de Ajuste Polinomial. Son descritas en detalle.

La motivación de estos problemas es la de mostrar el comportamiento de los algoritmos en problemas reales de cierta utilidad.

Las funciones de tests tienen las ventajas de ser ampliamente conocidas en la literatura y su gran sencillez. De forma complementaria a éstas, utilizamos tres problemas de optimización reales, que presentan un mayor comportamiento (y, por tanto, mayor dificultad):

*Sistema de Ecuaciones Lineales*[EMS97] Este problema puede ser enunciado como, dado la matriz  $A$  y el vector  $B$ , encontrar el vector  $X$  que cumpla la expresión  $A \cdot X = B$ . Por tanto, la función de evaluación es la medida del error:

$$P_{sle}(x_1, \dots, x_n) = \sum_{i=1}^n \sum_{j=1}^n (a_{ij} \cdot x_j) - b_j$$

Existe una dependencia entre los parámetros (no linealidad) que no se deteriora conforme aumenta el número de parámetros, que hace que este problema posea una alta dificultad.

En concreto, hemos considerado una instancia de dimensión 10 del problema, definida en la Figura 2.3.

Nombre	F(x)
Esfera [dJ75]	$\sum_{i=1}^n x_i^2$
Schwefel [Sch81]	$\sum_{i=1}^n \sum_{j=1}^i x_j^2$
Generalizada de Rastrigin [B92, TA89]	$a \cdot n + \sum_{i=1}^n -a \cdot \cos(\omega \cdot x_i)$ , $a = 10$ , $\omega = 2 \cdot \pi$
Griewangk	$\frac{1}{d} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ , $d = 4000$
Expansión de F10 [WBG95]	$f_{10}(x_1, x_2) + \dots + f_{10}(x_{i-1}, x_i) + \dots + f_{10}(x_n, x_1)$ donde $f_{10}(x, y) = (x^2 + y^2)^{0,25} [\sin(50 \cdot (x^2 + y^2)^{0,1}) + 1]$
Rosenbrock Generalizada [dJ75]	$\sum_i i = 1^{n-1} (100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$
Ackley [Ack87]	$20 + e - 20e^{-0,2\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}} - e \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)$
Bohachevsky [RC97]	$x_1^2 + 2x_2^2 - 0,3 \cdot \cos(3\pi x_1) \cdot \cos(4\pi x_2) + 0,3$
Watson [RC97]	$\sum_{i=1}^{30} \left( \sum_{j=1}^n j \cdot \left(\frac{i-1}{29}\right)^{j-1} \cdot x_{j-1} - \sum_{j=1}^n \left(\frac{i-1}{29}\right)^{j-1} \cdot x_j - 1 \right)^2$
Colville [Col68, RC97]	$100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10,1((x_2 - 1)^2 + (x_4 - 1)^2) + 19,8(x_2 - 1)(x_4 - 1)$

Tabla 2.1: Definición de las Funciones de Tests

Nombre	Rango	Dimensión
Esfera [dJ75]	$-2,56 \leq x_i \leq 5,12$	n=25
Schwefel [Sch81]	$-35 \leq x_i \leq 95$	n=25
Generalizada de Rastrigin [B92, TA89]	$-8,5 \leq x_i \leq 1,5$	n=25
Griewangk	$-200 \leq x_i \leq 1000$	n=25
Expansión de F10 [WBG95]	$-80 \leq x_i \leq 120$	n=25
Rosenbrock Generalizada [dJ75]	$-5,12 \leq x_i \leq 5,12$	n=25
Ackley [Ack87]	$-15 \leq x_i \leq 30$	n=25
Bohachevsky [RC97]	$-5,5 \leq x_i \leq 6,5$	n=2
Watson [RC97]	$-2 \leq x_i \leq 2$	n=6
Colville [Col68, RC97]	$-10 \leq x_i \leq 10$	n=4

Tabla 2.2: Dominios de las Funciones de Tests

$$\begin{array}{c}
 \left| \begin{array}{cccccccccc}
 5 & 4 & 5 & 2 & 9 & 5 & 4 & 2 & 3 & 1 \\
 9 & 7 & 1 & 1 & 7 & 2 & 2 & 6 & 6 & 9 \\
 3 & 1 & 8 & 6 & 9 & 7 & 4 & 2 & 1 & 6 \\
 8 & 3 & 7 & 3 & 7 & 5 & 3 & 9 & 9 & 5 \\
 9 & 5 & 1 & 6 & 3 & 4 & 2 & 3 & 3 & 9 \\
 1 & 2 & 3 & 1 & 7 & 6 & 6 & 3 & 3 & 3 \\
 1 & 5 & 7 & 8 & 1 & 4 & 7 & 8 & 4 & 8 \\
 9 & 3 & 8 & 6 & 3 & 4 & 7 & 1 & 8 & 1 \\
 8 & 2 & 8 & 5 & 3 & 8 & 7 & 2 & 7 & 5 \\
 2 & 1 & 2 & 2 & 9 & 8 & 7 & 4 & 4 & 1
 \end{array} \right| = \left| \begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \right| \left| \begin{array}{c} 40 \\ 50 \\ 47 \\ 59 \\ 45 \\ 35 \\ 53 \\ 50 \\ 55 \\ 40 \end{array} \right|
 \end{array}$$

Tabla 2.3: Matrices del Sistema de Ecuaciones Lineales

*Problema de Identificación de parámetros sonoros de modulación de frecuencia [TF93]* El problema se basa en especificar seis parámetros  $a_1, w_1, a_2, w_2, a_3, w_3$  del modelo de modulación sonora de frecuencias representada por

$$y(t) = a_1 \cdot \sin(w_1 \cdot t \cdot \theta + a_2 \cdot \sin(w_2 \cdot t \cdot \theta + a_3 \cdot \sin(w_3 \cdot t \cdot \theta))),$$

con  $\theta = \frac{2 \cdot \pi}{100}$ . La función objetivo es definida como la sumatoria del cuadrado de los errores entre los datos implicados y los del modelo:

$$P_{fms}(a_1, w_1, a_2, w_2, a_3, w_3) = \sum_{t=0}^{100} (y(t) - y_0(t))^2,$$

donde los datos del modelo se definen según la siguiente expresión:

$$y_0(t) = 1,0 \cdot \sin(5,0 \cdot t \cdot \theta - 1,5 \cdot \sin(4,8 \cdot t \cdot \theta + 2,0 \cdot \sin(4,9 \cdot t \cdot \theta))).$$

Cada parámetro toma valores en el rango  $[-6.4, 6.35]$ . Este problema es un problema multimodal muy complejo que presenta una fuerte dependencia entre valores.

*Problema de Ajuste Polinomial [SP95]* En este problema, el objetivo es encontrar los coeficientes  $z$  del siguiente polinomio:

$$P(z) = \sum_{j=0}^{2k} c_j \times z^j, \quad k > 0 \text{ is integer,}$$

en donde

$$P(z) \in [-1, 1], \text{ for } z \in [-1, 1], \text{ y}$$

$$P(1,2) \geq T_{2k}(1,2) \text{ y } P(-1,2) \geq T_{2k}(-1,2),$$

y  $T_{2k}(z)$  es un polinomio Chebychev de grado  $2k$ .

La solución a este problema depende de los coeficientes de  $T_{2k}(z)$ . Este polinomio oscila entre  $-1$  y  $1$  mientras su argumento  $z$  toma valores entre  $-1$  y

1. Fuera de esta región el polinomio asciende lentamente hacia valores positivos muy altos. Este problema tiene su origen en el diseño de filtros electrónicos y el proceso de optimización supone forzar la búsqueda de valores de parámetros con magnitudes muy distintas, algo muy común en los sistemas técnicos.

El polinomio Chebychev concreto empleado aquí es un problema de nueve parámetros:

$$T_8(z) = 1 - 32 \cdot z^2 + 160 \cdot z^4 - 256 \cdot z^6 + 128 \cdot z^8.$$

El seudocódigo siguiente es empleado para transformar las restricciones del problema en una función objetivo a minimizar, llamado  $P_{Chev}$ . Consideramos que  $C = (c_0, \dots, c_8)$  es la solución que se desea evaluar y  $P_C(z) = \sum_{j=0}^8 c_j \times z^j$ .

```

Choose  $p_0, p_2, \dots, p_{100}$  from  $[-1, 1]$ ;
 $R = 0$ ;
For  $i = 0, \dots, 100$  do
  If  $(-1 > P_C(p_i)$  or  $P_C(p_i) > 1)$  then  $R \leftarrow R + (1 - P_C(p_i))^2$ ;
If  $(P_C(1,2) - T_8(1,2) < 0)$  then  $R \leftarrow R + (P_C(1,2) - T_8(1,2))^2$ ;
If  $(P_C(-1,2) - T_8(-1,2) < 0)$  then  $R \leftarrow R + (P_C(-1,2) - T_8(-1,2))^2$ ;
Return  $R$ ;

```

Cada parámetro (coeficiente) está en el rango  $[-512, 512]$ , y el valor de la función objetivo es  $P_{Chev}(C^*) = 0$ .

**Condiciones de la experimentación.** Los experimentos han sido realizados bajo las siguientes condiciones:

- Los algoritmos son ejecutados 50 veces sobre cada función. Dado a su comportamiento no determinístico, el resultado de aplicar un algoritmo variará entre ejecuciones. El realizar un amplio número de evaluaciones (50) y considerar la media de ellas, permite obtener un valor robusto para compararlas entre sí.
- Cada ejecución termina en cuanto el algoritmo haya evaluado 1000000 soluciones durante la búsqueda. Este valor total incluye tanto las evaluaciones

del AE como las debidas a la BL. Esto permite ver si la mejora introducida por la BL compensa su coste (la disminución de evaluaciones del AE).

### 2.3.1.1. Características comunes de los distintos algoritmos

Las características comunes de los distintos algoritmos que se presentan a continuación:

- Se han utilizado los operadores de cruce aritmético (denotados ARIT) y  $BLX - \alpha$  con valor de  $\alpha$  0.3 y 0.5 (denotados  $BLX-0.3$  y  $BLX-0.5$ ), para todos los AEs excepto el CLPSO (que por definición no usa operador de combinación).
- Se ha aplicado con probabilidad 0.125 los operadores de mutación BGA y uniforme para los AGs (en el resto no se aplica mutación por definición).
- Se aplica la BL con distinta intensidad de BL ( $n_{int}$ ). Los valores de  $n_{int}$  aplicados son 0 (sin BL), 10, 50 y 100.
- Sobre los AGs se aplican los distintos métodos de selección comentados en la Sección 2.2.5.4.
- Sobre los AGEs se aplican las distintas estrategias de reemplazo comentadas en la Sección 2.2.5.5.
- Tamaños de la población considerada:
  - 30 para CLPSO, por ser un valor recomendado por sus autores [LQSB03].
  - 50 para CHC.
  - 60 para el resto.
- Si  $n_{int} > 0$  se aplica la BL al 6.25 % de los nuevos individuos.

### 2.3.1.2. Tests Estadísticos

No existe un procedimiento establecido y aceptado para comparar algoritmos sobre múltiples problemas. Un importante motivo es el comportamiento no determinístico de éstos, por lo que la diferencia detectada entre los resultados de dos algoritmos podría deberse a factores aleatorios, y no a una mejora real.

Para intentar determinar si las diferencias encontradas entre dos algoritmos son significativas (reales) o aleatorias, los investigadores pueden aplicar distintas técnicas estadísticas. Existen, principalmente, dos grupos de tests estadísticos:

- Los tests paramétricos, los más empleados. Se caracterizan en que utilizan por cada algoritmo y función el error medio alcanzado de un conjunto de ejecuciones y lo utilizan para identificar si la diferencia entre dos algoritmos es estadísticamente significativa. Un ejemplo de estos tests son el test de t-Student[Gos08] (para comparar dos algoritmos entre sí) y ANOVA (para comparar múltiples algoritmos entre sí).

Para poder aplicar dichos tests paramétricos con un mínimo de fiabilidad, los resultados de los algoritmos deben de seguir una distribución normal, y mantener una varianza homogénea. Cuando estas características son satisfechas los análisis paramétricos son correctos y seguros, y más sensibles (perceptivos) que los de la siguiente categoría, los tests no paramétricos.

- Los tests no-paramétricos, que utilizan una representación ordinal de los valores basada en el orden de los algoritmos para cada uno de los problemas. Un ejemplo de estos tests son el test de Wilcoxon (para comparar los resultados de dos algoritmos entre sí) y el test de Bonferroni-Dunn o el test de Holm (para comparaciones de varios entre sí).

A lo largo de esta memoria aplicaremos normalmente los tests no paramétricos, ya que es frecuente que las funciones de tests (las de este capítulo y las de otras) empleadas no cumplan las condiciones requeridas para poder usar de forma correcta comparaciones paramétricas [GMLH07, Dem06].

Usaremos un conjunto pequeño de test no-paramétricos robustos y seguros para realizar las comparaciones estadísticas:

- El test de Iman-Davenport. Es una variación del test de Friedman [Fri40], que es equivalente al conocido ANOVA, que nos puede permitir identificar si, a partir de los resultados de una serie de algoritmos (para una serie de funciones), existe al menor una diferencia significativa entre ellos.

Friedman ordena para cada función los distintos algoritmos en función de su valor medio para la función objetivo (de menor a mayor), asignándole un valor de posición ( $r_j$  para el algoritmo  $j$ ,  $r_j \in [1, \dots, k]$ , donde  $k$  es el

número total de algoritmos). Luego, calcula el valor de Friedman *Friedman* mediante la Ecuación 2.7.

$$\chi_F^2 = \frac{12N}{k \cdot (k+1)} \left[ \sum jR_j^2 - \frac{k \cdot (k+1)^2}{4} \right] \quad (2.7)$$

Como todo test estadístico, se guía sobre una hipótesis nula, que en este caso es la hipótesis de que los resultados de todos los algoritmos son equivalentes. Cuando esta condición se cumple cierta, el valor estadístico del test de Friedman sigue una distribución  $\chi_F^2$  con  $k - 1$  grados de libertad, siempre que  $N$  y  $k$  sean suficientemente grandes ( $N > 10$  y  $k > 5$ ).

Aunque podríamos haber aplicado el test de Friedman tal y como hemos definido, hemos aplicado finalmente una variante suya, conocida como el test de Iman-Davenport. El motivo es que sus autores demostraron que la función  $\chi_F^2$  de Friedman era excesivamente conservativa y definieron una variante con un mejor comportamiento, indicada en la Ecuación 2.8.

$$\text{Valor Iman-Davenport} = \frac{(N-1)\chi_F^2}{N \cdot (k-1) - \chi_F^2} \quad (2.8)$$

Bajo las mismas condiciones de equivalencia, el valor de Iman-Davenport debería de seguir una distribución  $F$  con  $k - 1$  y  $(k - 1) \cdot (N - 1)$  grados de libertad. Por tanto, si el valor del test de Iman-Davenport es mayor que el valor correspondiente de la distribución  $F$ , entonces esto quiere decir que existe al menos una diferencia estadísticamente relevante entre los algoritmos considerados.

El valor de  $F$  de referencia es el correspondiente a la distribución  $F$  con un parámetro de control (ver tabla B.4 en [Zar99]). En nuestros estudios comparativos aplicamos el valor  $F_F$  correspondiente a un 0.05 como parámetro de control (margen de error de un 5%).

Visualmente, este valor se muestra mediante tablas como la Tabla 2.4, que refleja un hipotético ejemplo con 5 algoritmos. En dicha tabla se muestra el valor Iman-Davenport, el valor  $F_F$  para el correspondiente grado de libertad, y si la hipótesis nula se acepta o rechaza. Cuando se rechaza, se ha detectado una diferencia significativa.

Como el valor del test de Iman-Davenport es mayor que  $F_F$  se rechaza la hipótesis nula (que todos los algoritmos son equivalentes), obteniendo, por

Nombre	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
Ejemplo	10.149	2.46	R

Tabla 2.4: Ejemplo de test Iman-Davenport

tanto, que existe al menos una diferencia significativa entre los algoritmos del ejemplo.

El test de Iman-Davenport sólo indica que existe alguna diferencia significativa, no cual o cuales algoritmos ofrecen diferencias estadísticamente relevantes. Para eso, existe una serie de test *post-hoc* que pueden ser aplicados en ese caso para identificar entre qué algoritmos existe una diferencia relevante. Estos tests son los dos siguientes.

- El test de Bonferroni-Dunn. Es un test *post-hoc* muy similar al test de Turkey para ANOVA y es de aplicación cuando se desea compara un algoritmo contra otros. Requiere haber aplicado anteriormente el test de Friedman (o equivalente, como el test de Iman-Davenport) que confirme que exista una diferencia estadísticamente significativa. Utiliza el orden medio de cada algoritmo. Para calcular el orden medio de un algoritmo, se ordenan función por función los distintos algoritmos por su valor para la función objetivo (primero aquellos con mejor valor), y para cada algoritmo se calcula la media de sus posiciones para las distintas funciones.

Este test devuelve un valor, llamado *diferencia crítica* (CD). Cuando entre dos algoritmos la diferencia entre sus respectivos órdenes medios es mayor que dicho valor, se considera que la diferencia entre ellos es estadísticamente significativa. Se calcula mediante la Ecuación 2.9.

$$CD = q_\alpha \sqrt{\frac{k \cdot (k + 1)}{6 \cdot N}} \quad (2.9)$$

Donde  $q_\alpha$  es el valor crítico de  $Q'$  para una comparación múltiple con una variable de control (tabla B.16 en [Zar99]). En nuestros experimentos elegiremos el valor de diferencia crítica correspondiente para un margen de error del 5%.

El test de Bonferroni-Dunn permite determinar que un algoritmo A es significativamente mejor que otro B si  $Rango_A + CD < Rango_B$ . En nuestro

caso, lo aplicaremos principalmente para identificar aquellos algoritmos que sean significativamente peor que el mejor algoritmo (el algoritmo con menor orden medio).

Visualmente mostramos este test mediante un diagrama de columnas. Cada columna representa un algoritmo y su altura es el orden medio que éste obtiene. El valor  $CD + Menor Orden Medio$  es obtenido y representado mediante una línea horizontal que divide a los algoritmos en dos grupos excluyentes: Aquellos cuyos valores están por encima de la línea horizontal, identificados como estadísticamente peores que el mejor; y aquellos cuyo valor es inferior a dicha línea, identificándose como estadísticamente equivalentes al mejor.

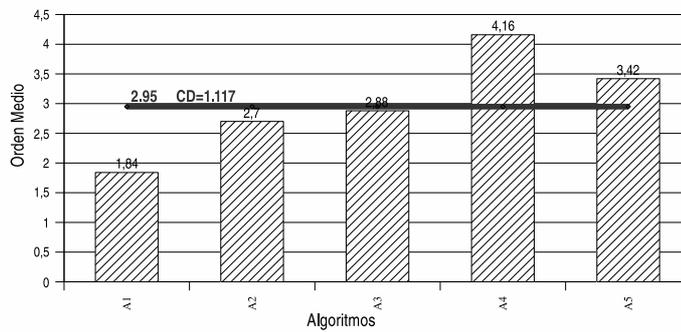


Figura 2.13: Ejemplo del test de Bonferroni-Dunn

Siguiendo el ejemplo hipotético anterior, la Figura 2.13 muestra el resultado del test de Bonferroni-Dunn. Dicha figura muestra que el algoritmo con menor orden medio es A1, y que tanto A4 como A5 son estadísticamente peores que A1. No dice nada respecto a la relación entre A1, A2 y A3.

- El test de Holm [Hol79]. Éste, al igual que el test de Bonferroni-Dunn, es un test *post-hoc* que requiere que el de Iman-Davenport identifique una diferencia significativa para poder aplicarse.

Este test identifica al algoritmo con menor orden medio, y compara al resto con éste. El estadístico empleado para comparar el algoritmo  $i$ -ésimo con el  $j$ -ésimo, denominado  $z$  se calcula mediante la Ecuación 2.10.

$$z = (R_i - R_j) / \sqrt{\frac{k \cdot (k + 1)}{6 \cdot N}} \quad (2.10)$$

Una vez encontrado el valor  $z$  se encuentra la probabilidad correspondiente siguiendo una distribución normal ( $p$ ). Una vez obtenida para cada algoritmo su probabilidad  $p_i$  se ordenan los algoritmos de tal forma que  $p_1 \leq p_2 \leq p_3 \leq \dots \leq p$ .

El método de Holm compara luego cada  $p_i$  con  $\alpha/(k-i)$  comenzando desde el valor más significativo. Si  $p_1$  es menor que  $\alpha/(k-1)$ , la correspondiente hipótesis se rechaza (y por tanto, existe una diferencia significativa) con un margen de error de  $\alpha$ . Pasaríamos en ese caso a comparar si  $p_2$  es menor que  $\alpha/(k-2)$ , y si la segunda hipótesis se rechaza el proceso continúa. En cuanto una determinada hipótesis no puede ser rechazada, todas las restantes se mantienen como aceptadas (y por tanto, se considera tanto a ese algoritmo como a los restantes por comparar como estadísticamente equivalentes al mejor). En nuestros estudios comparativos aplicaremos el test de Holm con  $\alpha=0.05$ , garantizando así un margen de error del 5%.

El test de Holm presenta como ventajas respecto al de Bonferroni-Dunn que es más potente, y no hace ninguna suposición adicional sobre las hipótesis chequeadas.

Para mostrar los resultados del tests, emplearemos una tabla específica. Esta tabla se compone de 6 columnas y una fila por algoritmo comparado con el algoritmo de referencia. Las dos primeras columnas identifican el orden y el nombre del algoritmo. La columna identificada como  $z$  es el valor obtenido en la Ecuación 2.10. La siguiente es el valor  $p$  asociado al  $z$  correspondiente a partir de una distribución normal, y a continuación el valor de  $\alpha/i$  que se ha de comparar.

Finalmente, la última columna muestra si se rechaza (R) la hipótesis asociada a la fila o se acepta (A).

Dado que los algoritmos están ordenados de peor a mejor, se dividirá mediante una línea horizontal los algoritmos en dos grupos. Aquellos que estén por encima de dicha línea son estadísticamente peores que el de referencia, mientras que aquellos situados por debajo de la línea serían equivalentes (sin diferencia significativa detectada).

La Tabla 2.5 muestra el resultado de aplicar el test de Holm sobre el ejemplo anterior.

i	Algoritmo	z	p	$\alpha/i$	Aceptada o Rechazada
4	A4	5.187	2.129-07	0.0125	R
3	A5	3.532	0.000410	0.0167	R
2	A3	2.325	0.020044	0.0250	R
1	A2	1.923	0.054477	0.0500	A

Tabla 2.5: Test de Holm sobre el algoritmo A1

De dicha tabla puede observarse que identifica a los algoritmos A3, A4 y A5 como estadísticamente peores que A1. No sólo confirma los resultados indicados por el test de Bonferroni-Dunn, sino que en ocasiones puede identificar como estadísticamente peores a más algoritmos que éste, debido a una mayor sensibilidad. En este caso, el test de Holm identifica a A3 como peor que A1, aún cuando el de Bonferroni-Dunn no era capaz de detectar dicha diferencia significativa.

- El test de Wilcoxon.

Es un test no paramétrico empleado para comparar dos algoritmos entre sí. Permite identificar tanto si existe una diferencia significativa entre dos algoritmos para un conjunto de funciones como determinar cuál de ellos es mejor.

El procedimiento que aplica es el siguiente:

1. Calcula para cada función la diferencia entre los valores obtenidos por el algoritmo a comparar y el de referencia.
2. Se dividen las funciones en dos grupos: Funciones para las que el algoritmo de referencia es el mejor, y funciones para los que el algoritmo de referencia ofrece el peor resultado. Aquellas funciones en donde ambos algoritmos obtengan el mismo resultado se dividirán por igual entre ambos grupos (la mitad como positivas y la otra como negativas).
3. Se utilizan los valores absolutos de las diferencias para crear un orden de funciones: se asigna el menor valor a la diferencia con menor valor absoluto.
4. Se acumulan los valores de orden para cada uno de los grupos anteriores (*Ranking Positivo*,  $R^+$ , para el primer grupo, y *Ranking Negativo*,  $R^-$

para el segundo).

5. Se calcula  $T = \min(R^+, R^-)$ .
6. Se consulta el valor de la distribución T de Wilcoxon para  $N$  grados de libertad (Tabla B.123 en [Zar99]) como valor de referencia.
7. Se compara el valor  $T$  con el de referencia. Cuando  $T < referencia$ , se rechaza la hipótesis nula (es decir, la diferencia es estadísticamente significativa).
8. Si se identifica una diferencia relevante entre el algoritmo a comparar y el de referencia, se identifica aquel que presenta mejor comportamiento:
  - Si  $T = R^+$  ( $R^+ < R^-$ ), es mejor el algoritmo comparado.
  - Si  $T = R^-$  ( $R^+ > R^-$ ), es mejor el algoritmo de referencia.

Los resultados de este test los mostramos mediante tabla utilizando un algoritmo de referencia. En dicha tabla, las filas representan los distintos algoritmos. Las columnas muestran cada uno de los siguientes valores:

- El *Ranking Positivo*,  $R^+$ . Cuando  $T = R^+$ , se le añade ( $T$ ) junto al valor, para indicar que éste es el valor empleado para aceptar o rechazar la hipótesis nula. En este caso, el algoritmo de referencia es peor que el comparado.
- El *Ranking Negativo*,  $R^-$ . Cuando  $T = R^-$  se le añade ( $T$ ) junto al valor, para indicar que éste es el valor empleado para aceptar o rechazar la hipótesis nula. En este caso, el algoritmo de referencia es mejor que el comparado.
- El valor de referencia (Tabla B.12 de [Zar99]). Es el valor que nos permite aceptar o rechazar la hipótesis nula. Cuando uno de los *ranking* anteriores es menor que dicho valor, se rechaza la hipótesis nula.
- Indicación de si se acepta (A) o rechaza (R) la hipótesis nula.

La posición de '(T)' indica cómo de bueno es el algoritmo de referencia (o propuesta) frente al resto. Cuando aparece el '(T)' en la columna de la derecha, junto a  $R^-$ , el algoritmo de referencia presenta mejores resultados que el algoritmo comparado, mientras que si aparece a la izquierda, el algoritmo de referencia ofrece peores resultados.

En nuestro ejemplo, la Tabla 2.6 muestra los resultados de aplicar Wilcoxon.

Algoritmo	R+	R-	Referencia	Aceptada o Rechazada
A2	209.0	116.0 (T)	89	A
A3	279.5	45.5 (T)	89	R
A4	301.0	24.0 (T)	89	R
A5	305.0	19.5 (T)	89	R

Tabla 2.6: Ejemplo del test de Wilcoxon Test usando A1 como referencia

Se puede observar que el test de Wilcoxon cataloga A1 como mejor que A3, A4 y A5, pero no detecta ninguna diferencia destacable entre A1 y A2 (porque en ese caso el valor  $T$ , 116, no es menor que el valor de referencia 89).

A continuación resumimos el proceso seguido para identificar si hay uno o varios que sean estadísticamente mejores que otros:

1. Aplicar el test de Iman-Davenport para confirmar que exista al menos una diferencia significativa, ya que en ese caso, se pueden aplicar los test *post-hoc* (el test de Bonferroni-Dunn y el Holm).
2. Si el test Iman-Davenport no detecta diferencia significativa, se termina la comparación (o se aplica el test de Wilcoxon).
3. En otro caso podemos aplicar el test de Bonferroni-Dunn o el de Holm para comparar los distintos algoritmos con aquél con mejor orden medio. Nos permiten identificar aquellos algoritmos que son estadísticamente peores que aquél con menor orden medio.

Dado que el test de Holm ordena los algoritmos por bondad, mientras que con el de Bonferroni-Dunn se pueden mostrar los resultados en un orden deseado, aplicaremos este último para mostrar tendencias de los algoritmos al modificar un parámetro, de esta forma se puede observar más fácilmente cómo afecta (si los resultados mejoran o empeoran conforme se varía dicho parámetro). Si simplemente nos interesa una ordenación de algoritmos, aplicamos con mayor frecuencia el test de Holm, por su mayor potencia.

En el caso de que tengamos una propuesta y queramos compararla con un conjunto de algoritmos, podremos aplicar el test de Wilcoxon directamente, ya que este test no requiere el análisis anterior del test de Friedman o del de Iman-Davenport.

### 2.3.1.3. Otras Medidas

Todos los tests no paramétricos empleados consideran únicamente la posición relativa de cada algoritmo, y no los valores de error concretos (o su valor para la función objetivo). Dado que ésta es una información muy interesante, adicionalmente al empleo de los comparadores no-paramétricos, calcularemos el error acumulado (EAc) de cada algoritmo. El EAc es una medida que combina los diferentes errores relativos para cada función, y que permite identificar los algoritmos más robustos. El EAc se calcula mediante las Ecuaciones 2.11, 2.12 y 2.13.

$$EAc(Alg_i) = \sum_{f=1}^{N_{Fun}} RE(Alg_i, fun) \quad (2.11)$$

$$RE(Alg_i, fun) = \begin{cases} 0 & \text{si } \nabla E_{fun} = 0 \\ \frac{E(Alg_i, fun) - Min_{fun}}{\nabla E_{fun}} & \text{si } \nabla E_{fun} > 0 \end{cases} \quad (2.12)$$

$$\nabla E_{fun} = \max(0, E_{Max}(fun) - Umbral(fun)) \quad (2.13)$$

en donde  $Umbral(fun) = 0$ ,  $E_{Max}(fun) \geq E(Alg_i, fun) \forall fun = 1, \dots, N_{Fun}$ ,  $\forall i = 1, \dots, NumAlgoritmos$ .

Este valor es mostrado gráficamente utilizando un diagrama de barra en el que las columnas son los algoritmos comparados y las alturas son los EAc de cada algoritmo.

La Figura 2.14 muestra los errores acumulados del ejemplo anterior. Muestra cómo A1 posee el menor error acumulado, y no existe diferencias apreciables en los errores de A2 y A3.

### 2.3.2. Estudio Individual de cada AE considerado

El comportamiento de cada AE está determinado por una serie de operadores (cruce, mutación, selección, reemplazo, ...). Algunos de estos operadores están definidos en la especificación del propio algoritmo, son característicos e inalterables, pero otros pueden ser elegidos de entre una amplia variedad, es decir, son adaptables. Esto conduce a que por cada modelo puedan definirse muchas variantes. El número de variantes depende del AE considerado, en función del conjunto distintos operadores adaptables consideremos (explicados en 2.2.5).

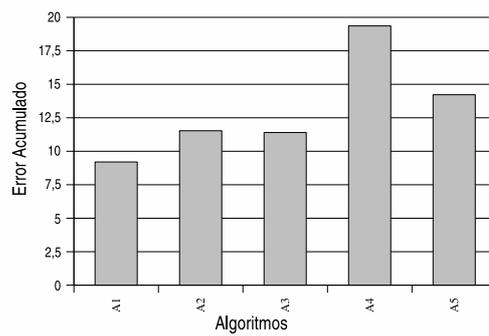


Figura 2.14: Ejemplo de Errores Acumulados

Esta flexibilidad ha hecho necesaria realizar, para cada AE, un estudio comparando entre sí sus distintas variantes. El objeto de este estudio es doble:

- Por un lado, determinar el mejor modelo que use dicho AE.

A la hora de determinar el AE más recomendable identificaremos para cada AE las variantes que presenten un mejor comportamiento para una posterior comparación final.

- Por otro lado, estudiar cómo influye la incorporación de la BL a cada uno de los algoritmos considerados.

Dentro del contexto de los AMs, no sólo nos interesa elegir el modelo más adecuado, sino también su comportamiento en función de su uso combinado con la BL. Para estudiar cómo se comporta cada AE (o variante) cuando se le incorpora la BL, para los dos tipos de BL considerados (Solis Wets y Simplex) estudiamos de forma separados las variantes para los valores de intensidad  $n_{int}=0$  (sin BL), 10, 50, 100.

Una vez obtenidos los resultados nos centramos en señalar principalmente tres aspectos:

- *Influencia de la BL.* El principal objetivo del capítulo es estudiar la conveniencia de usar un AE u otro para construir un AM, por lo que es fundamental conocer la tendencia del algoritmo al aumentar la intensidad (si

mejora, empeora, o existe un valor de intensidad que suponga un equilibrio entre ambas tendencias).

- *Influencia de los operadores.* Dado que la elección del operador puede poseer mucha influencia en los resultados se van a comparar algoritmos que se diferencian en los operadores empleados. Comparando estos algoritmos entre sí se estudia la conveniencia de cada uno de esos operadores para cada valor de intensidad de la BL. De esa forma, se puede no sólo recomendar un esquema de AE, sino también la combinación de operadores que consideremos más óptima.
- *Mejores variantes.* Para poder comparar posteriormente los distintos AEs entre sí, dentro de cada análisis del AE, se elegirá un par de variantes que presenten los mejores resultados como representantes de ese AE en el análisis comparativo final. La elección de dichas variantes se han realizado siguiendo las distintas medidas utilizadas, y aplicando comparadores estadísticos no paramétricos. Sin embargo, en ciertos casos, la elección no es directa (por usarse más de un criterio de comparación).

Se han analizado los siguientes AEs:

- El algoritmo CHC. Se analiza el comportamiento del uso combinado del CHC con la BL para los distintos métodos de cruce.
- La BD. Se analiza el comportamiento del uso combinado de la BD con la BL para los distintos métodos de cruce.
- CLPSO. Se analiza el comportamiento del algoritmo CLPSO.
- AGG. Se analiza el comportamiento del uso combinado del AGG con la BL para cada uno de los métodos de selección (y para los distintos métodos de cruce).
- AGE. Se analiza el comportamiento del uso combinado del AGE con la BL para cada una de las combinaciones de los métodos de selección y las estrategias de reemplazo consideradas (explicadas en las Secciones 2.2.5.4 y 2.2.5.5).

### 2.3.2.1. Estudio Experimental del algoritmo CHC

En este apartado analizaremos el comportamiento de CHC para los distintos operadores de cruce, y su evolución conforme se le incrementa la intensidad de la BL (aumentando así la importancia de la BL en la búsqueda).

Hemos implementado varias instancias o variantes del CHC que se distinguen por el método de cruce, el método de BL y la intensidad de la BL empleada para dicho método ( $n_{int}$ ).

Las características de los algoritmos comparados son:

- Una población de 50 individuos.
- Los operadores de cruce aritmético y  $BLX - \alpha$  (con  $\alpha = 0,3$  y  $0,5$ ).
- Los valores de intensidad  $n_{int} = 0, 10, 500, 100$ .
- Los métodos de BL Simplex y Solis Wets.

Las variantes se identifican como “Cruce”, a las que no se le aplica la BL; y “Cruce BL  $n_{int}$ ” a las que sí. *Cruce* toma valores ARIT (Aritmético), BLX-0.3 y BLX-0.5; *BL* toma como valores SW (Solis Wets) y S (Simplex); y  $n_{int}$  toma los valores 10, 50 y 100. En total, son 21 variantes (9 por cada BL y 3 sin BL, una por cada cruce).

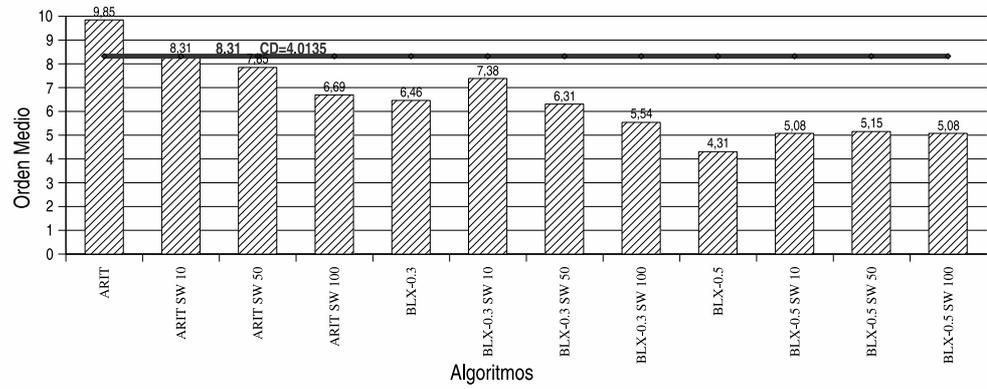
A la hora de estudiarlos, los hemos agrupado en 2 grupos de 10 variantes cada uno, uno por cada método de BL (la variante que no usa la BL es introducida como referencia en ambos).

BL	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
SW	3.016	1.86	R
S	5.398	1.86	R

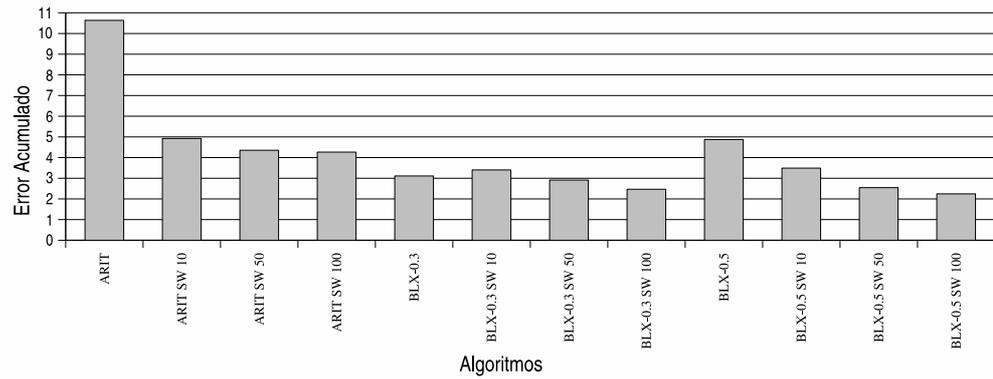
Tabla 2.7: Test de Iman-Davenport con CHC

En primer lugar, aplicamos el test de Iman-Davenport para determinar si existen diferencias significativas en cada uno de estos dos grupos. Dicho resultado, reflejado en la Tabla 2.7, muestra que existe en ambos diferencias significativas, por lo que es necesario un estudio más detallado.

Como test *post-hoc* empleamos el test de Bonferroni-Dunn porque nos permite mostrar mejor de forma gráfica la evolución de los algoritmos conforme aumenta la intensidad ( $n_{int}$ ). Para apoyar el análisis, calcularemos también los errores acumulados.



(a) Test de Bonferroni-Dunn



(b) Errores Acumulados

Figura 2.15: Comparación CHC Sin BL y Solis Wets

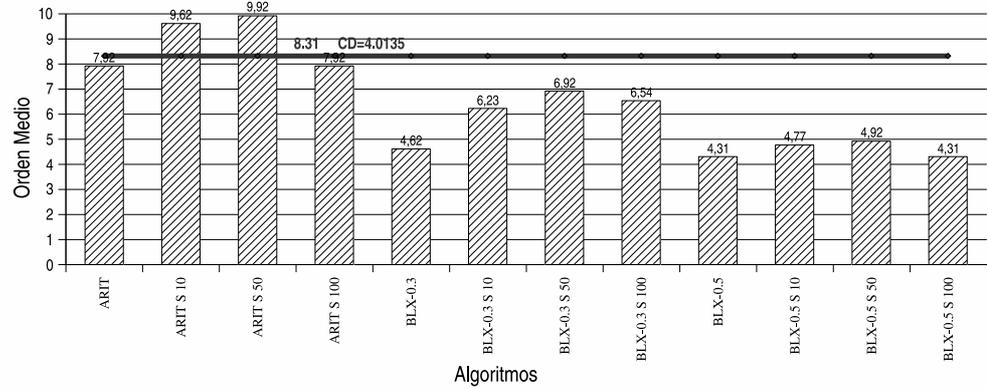
**CHC con Solis Wets** La Figura 2.15 muestra el resultado de comparar entre sí el CHC-SW con los distintos operadores de cruce, y distintas intensidades de BL, mostrando tanto el test de Bonferroni-Dunn como los EAc. De dicha figura

podemos extraer las siguientes conclusiones:

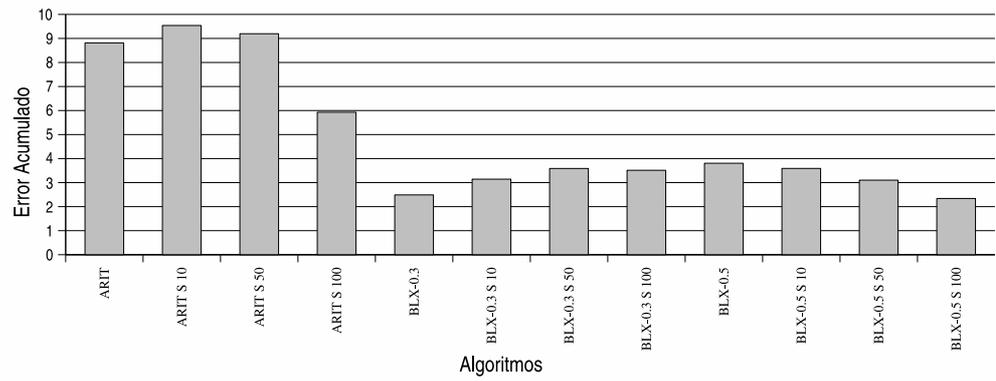
- *Influencia de la BL.* Puede observarse que la incorporación de mayor BL influye de forma distinta en función del operador de cruce empleado (aunque para todos puede generar una mejora significativa). Con el operador de cruce aritmético se obtienen mejores resultados cuanto mayor BL se emplee. Por contra, con *BLX-0.3* se observa que la incorporación de BL no siempre mejora los resultados, sino que es necesaria una cierta intensidad de BL para mejorar. Por otro lado, aunque según el test de Bonferroni-Dunn el incrementar la BL sobre el cruce *BLX-0.5* no parece influir en los resultados, se observa en EAc que sí se produce una mejora significativa en los resultados de ciertas funciones. Lo que sucede es que esa mejora se muestra insuficiente para cambiar las posiciones relativas de los algoritmos.
- *Influencia del operador de cruce.* Puede verse que el operador de cruce posee una influencia muy importante en los resultados, tanto en orden medio como en EAc. El operador aritmético presenta los peores resultados en orden medio, mientras que con el *BLX-0.3* y el *BLX-0.5*, aunque respecto al EAc ambos presentan resultados parecidos, es el *BLX-0.5* el que presenta el menor orden medio (y bastante independiente de la intensidad de BL aplicada).
- *Mejores variantes.* Se observa que las mejores variantes son los algoritmos con el operador de cruce *BLX- $\alpha$*  y alta intensidad de BL, como el *BLX-0.5 SW 50* y *BLX-0.5 SW 100*. Es destacable que la mejor variante identificada por cada diagrama es distinto: aunque el uso de *BLX-0.5* sin BL presenta muy buenos resultados respecto al orden medio, el EAc correspondiente es bastante alto, por lo que hemos elegido el empleo de alta intensidad.

**CHC con Simplex** La Figura 2.16 muestra el resultado de comparar entre sí CHC-S con los distintos operadores de cruce, y distintas intensidades de BL, mostrando tanto el test de Bonferroni-Dunn como los EAc. De dicha figura podemos extraer las siguientes conclusiones:

- *Influencia de la BL.* Hay una diferencia importante entre aplicar el Solis Wets y el Simplex. Mientras que con el Solis Wets la BL tendía a mejorar los resultados, el aumentar la intensidad con Simplex hace que los resultados empeoren, tanto en orden medio como en EAc (únicamente con *BLX-0.5*



(a) Test de Bonferroni-Dunn



(b) Errores Acumulados

Figura 2.16: Comparación CHC Sin BL y Simplex

el EAc disminuye conforme aumenta la BL). De este comportamiento se puede observar la poca conveniencia de aplicar el Simplex bajo CHC para el conjunto de casos de tests considerado.

- *Influencia del operador de cruce.* Dado que la BL no origina mejoras en los resultados, las diferencias entre los distintos métodos de cruce son más evidentes que con el Solis Wets (es destacable el alto EAc del aritmético). El operador de cruce *BLX-0.5* es el que ofrece los mejores resultados.
- *Mejores variantes.* Las mejores variantes son las que hacen uso del *BLX-0.5* y de la BL (*BLX-0.5 S 50* y *BLX-0.5 S 100*), que poseen los menores EAcs y están entre las 4 que poseen mejor orden medio.

**Elección de la Búsqueda Local (Solis Wets Versus Simplex)** En las secciones anteriores hemos comparado de forma separada los resultados las variantes meméticas del CHC con cada una de las BLs consideradas (Solis Wets y Simplex), sin entrar en valoraciones sobre la conveniencia de una sobre otra. Sin embargo, la elección de la BL, como el resto de decisiones, puede ser crucial.

En este apartado vamos a comparar entre sí las variantes meméticas (CHC-SW y CHC-S) entre sí, para intentar determinar el método de BL más recomendable. Dado que esto supone comparar entre sí 21 variantes, a la hora de compararlas las agrupamos por su operador de cruce (hay 7 por cada uno).

Cruce	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
ARIT	4.131818	2.23	R
BLX-0.3	3.764575	2.23	R
BLX-0.5	0.923077	2.23	A

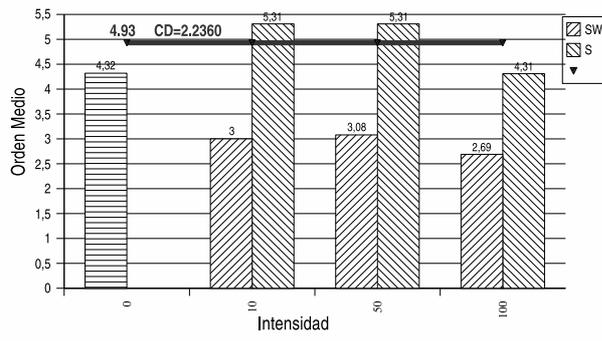
Tabla 2.8: Test de Iman-Davenport con CHC (SW y S)

En primer lugar, vamos a aplicar el test Iman-Davenport para cada uno de los grupos. La Tabla 2.8 muestra los resultados. De dicha tabla se observa que tanto para los métodos ARIT y BLX-0.3 (no así para BLX-0.5) existe una diferencia significativa. Lo cual justifica un mayor estudio.

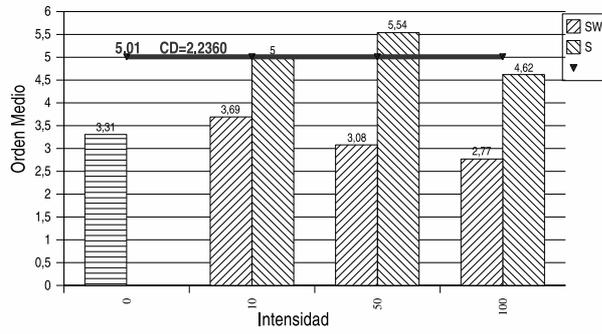
La Figura 2.17 muestra el resultado de aplicar el test de Bonferroni-Dunn sobre los grupos *ARIT* y *BLX-0.3*, y visualiza el orden medio para el grupo *BLX-0.5* (porque el test de Bonferroni-Dunn no es aplicable en este caso). De dicha

figura podemos observar que en la casi totalidad de los casos los resultados de las variantes con Solis Wets son superiores a las que hacen uso del Simplex. Sólo en un caso (con BLX-0.5 y  $N_{int} = 100$ ) el Simplex ofrece un resultado equivalente. Si bien es cierto que las diferencias se reducen conforme mayor diversidad aporta el cruce.

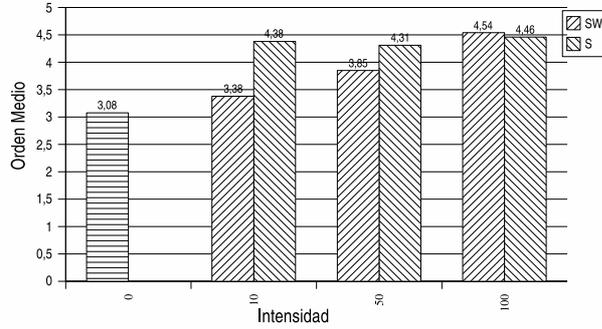
**Conclusiones** La variante memética del CHC con Solis Wets (CHC-SW) mejora a CHC, aunque para el Simplex sólo a partir de cierta intensidad (para el Solis Wets es mejor cuanto mayor sea la intensidad). Respecto al análisis de los operadores de cruce, es el operador de cruce BLX-0.5 con los que se obtienen los mejores resultados, aunque los resultados con el operador BLX-0.3 son bastante parecidos (no así usando el cruce aritmético, ya que éstos son significativamente peores). Respecto a la BL, es el Solis Wets el que ofrece mejores resultados. Por tanto, las mejores variantes son aquellas que usan el *BLX* - 0,5 y el *Solis Wets* con un valor alto de intensidad. (*CHC (BLX-0.5 SW 50)* y *CHC (BLX-0.5 SW 100)*).



(a) Cruce Aritmético



(b) Cruce BLX-0.3



(c) Cruce BLX-0.5

Figura 2.17: Orden Medio para CHC con Solis Wets y Simplex

### 2.3.2.2. Estudio Experimental de la Búsqueda Dispersa

En este apartado analizaremos el comportamiento de la BD para los distintos operadores de cruce, y su evolución conforme se le incrementa la intensidad de la BL (aumentando así la importancia de la BL en la búsqueda).

Hemos implementado varias instancias o variantes de la BD que se distinguen por el método de cruce, el método de BL y la intensidad de la BL empleada para dicho método ( $n_{int}$ ).

Las características de los algoritmos comparados son:

- Tamaño de la población:  $Psize = 100$ ,  $b_1 = 10$  y  $b_2 = 10$ .
- Una población de 50 individuos.
- Los operadores de cruce aritmético (*ARIT*) y  $BLX - \alpha$  (con  $\alpha=0.3$  y  $0.5$ ).
- Los valores de intensidad  $N_{int} = 0, 10, 500, 100$ .
- Los métodos de BL Simplex y Solis Wets.

Las variantes se identifican como “Cruce BL  $n_{int}$ ”, Denotamos “ $BLX - \alpha$ ” y “ARIT” a las variantes que no hacen uso del método de mejora. *Cruce* toma valores ARIT (Aritmético), BLX-0.3 y BLX-0.5. *BL* toma como valores SW (Solis Wets) y S (Simplex). El parámetro  $n_{int}$  toma los valores 10, 50 y 100. En total, son 19 variantes (9 por cada BL y 3 sin BL, una por cada cruce).

A la hora de estudiarlos los agrupamos en 2 grupos de 10 variantes cada uno, uno por cada método de BL (la variante que no usa la BL es introducida como referencia en cada uno).

BL	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
SW	3.727	1.86	R
S	2.654	1.86	R

Tabla 2.9: Test de Iman-Davenport con Búsqueda Dispersa

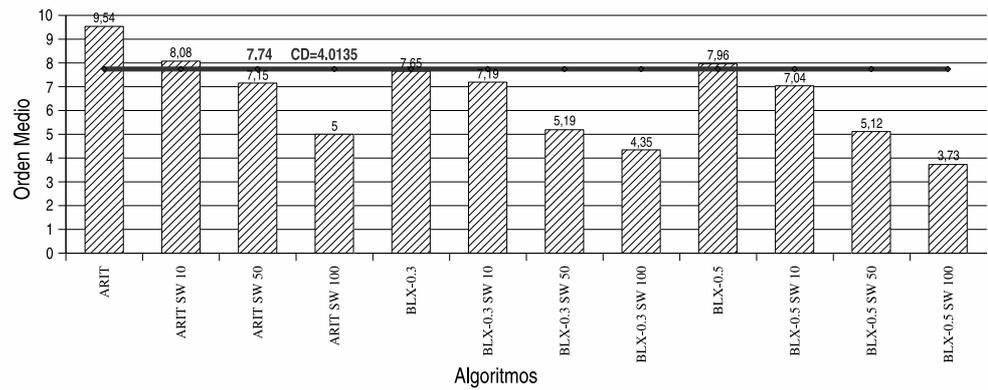
En primer lugar, aplicamos el test de Iman-Davenport para determinar si existen diferencias significativas en cada uno de estos dos grupos. De dicho resultado, reflejado en la Tabla 2.9, se muestra que existe en ambos casos diferencias significativas, por lo que es necesario un estudio más detallado.

Como test *post-hoc* emplearemos el test de Bonferroni-Dunn porque nos permite mostrar mejor de forma gráfica la evolución de los algoritmos conforme aumenta la intensidad ( $n_{int}$ ). Para apoyar el análisis, calcularemos también el EAc.

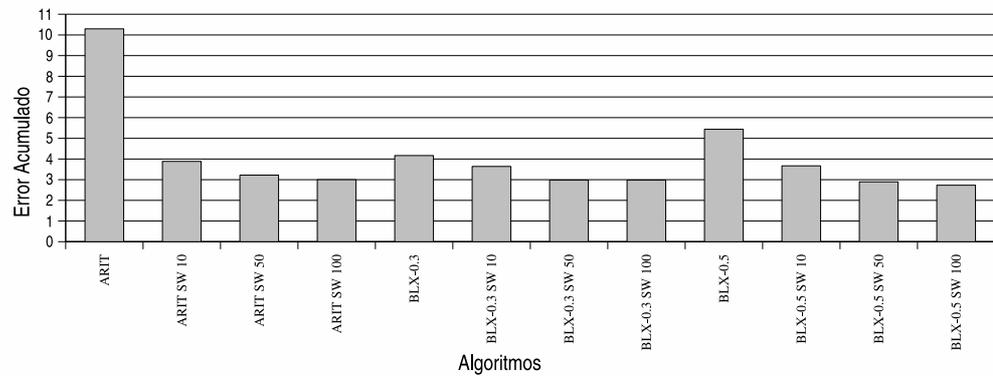
**BD con Solis Wets** La Figura 2.18 muestra el resultado de comparar entre sí BD-SW con los distintos operadores de cruce, y distintas intensidades de BL, mostrando tanto el test de Bonferroni-Dunn como los EAc. De dicha figura podemos extraer las siguientes conclusiones:

- *Influencia de la BL.* En la BD el incremento de la intensidad de la BL siempre una mejora en todos los casos, a diferencia de CHC.
  
- *Influencia del operador de cruce.* Al igual que con CHC, las variantes que hacen uso del cruce aritmético obtienen peores resultados que con el  $BLX - \alpha$ . Respecto al cruce  $BLX - \alpha$ , en la BD no se observa una diferencia significativa entre  $BLX-0.3$  y  $BLX-0.5$ .
  
- *Mejores variantes.*

La elección de la mejor variante es clara, el algoritmo con menor orden medio ( $BLX-0.5 SW 100$ ) posee a su vez el menor EAc. La otra elección realizada es el  $BLX-0.3 SW 100$  ya que es el siguiente que posee menor orden medio, y las diferencia de EAc es muy ligera respecto a aquel que posee el segundo menor EAc (el  $BLX-0.5 SW 50$ ).



(a) Test de Bonferroni-Dunn

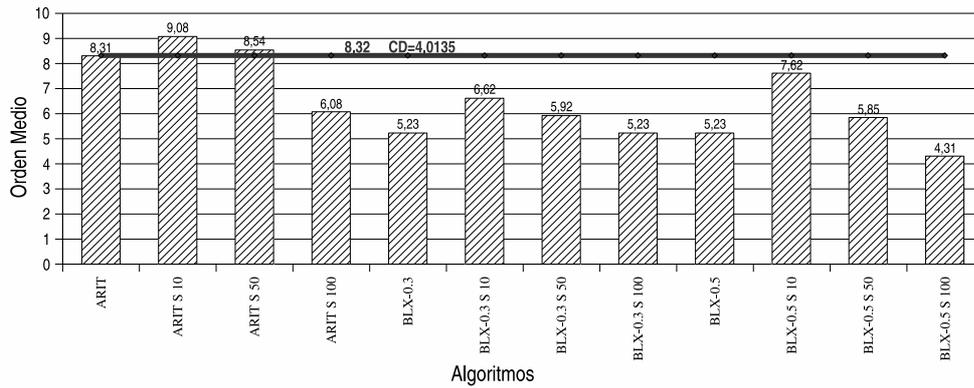


(b) Errores Acumulados

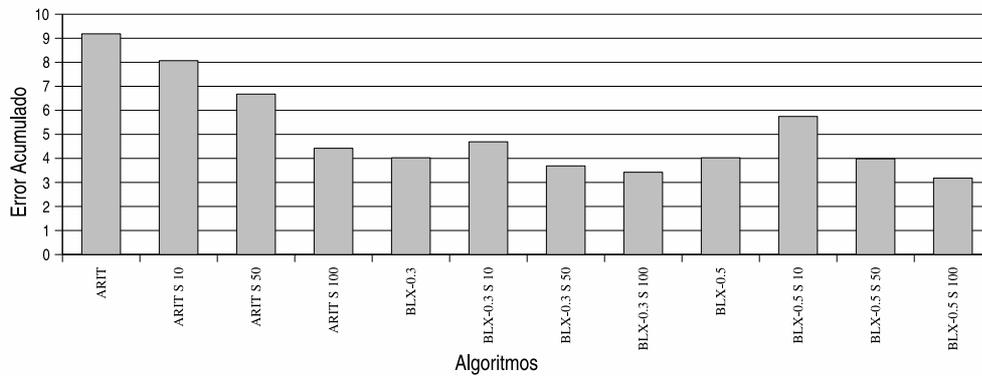
Figura 2.18: Comparación BD Sin BL y Solis Wets

**BD con Simplex** La Figura 2.19 muestra el resultado de comparar entre sí BD-S con los distintos operadores de cruce, y distintas intensidades de BL, mostrando tanto el test de Bonferroni-Dunn como los EAc. De dicha figura podemos extraer las siguientes conclusiones:

- Influencia de la BL.** Se observa que requiere un cierto valor de intensidad,  $n_{int}$ , a partir del cual la BL ofrece una mejora, a diferencia de cuando se emplea el Solis Wets. Esto parece indicar que el método Simplex requiere una



(a) Test de Bonferroni-Dunn



(b) Errores Acumulados

Figura 2.19: Comparación BD Sin BL y Simplex

mayor intensidad que el método Solis Wets para obtener buenos resultados.

- *Influencia del operador de cruce.* Al igual que pasada aplicando el método Solis Wets, el operador de cruce aritmético es claramente peor. Además, se observa una diferencia entre  $BLX-0.3$  y  $BLX-0.5$ . Para el mismo valor de intensidad parece que el uso del  $BLX-0.3$  presenta los mejores resultados con menor valor de intensidad (o sin BL), mientras que con una intensidad alta ( $n_{int} = 100$ ) es el  $BLX-0.5$  el que presenta mejor comportamiento. Esto puede deberse a que es necesario un equilibrio de exploración, y que

sólo cuando se aplica mucha explotación por medio de la BL, es cuando ofrece una mejora la mayor diversidad del *BLX-0.5*.

- *Mejores variantes.* Consideramos que *BLX-0.3 S 100* y *BLX-0.5 S 100* son las que ofrecen los mejores resultados (poseen tanto los menores EAcs como los menores órdenes medios).

**Elección de la BL (SW Versus S)** En este apartado vamos a comparar entre sí las variantes meméticas (con Solis Wets y con Simplex) entre sí, para intentar determinar el método de BL más recomendable. Dado que esto supone comparar entre sí 21 variantes, a la hora de compararlas las agrupamos por su operador de cruce (hay 7 por cada operador de cruce).

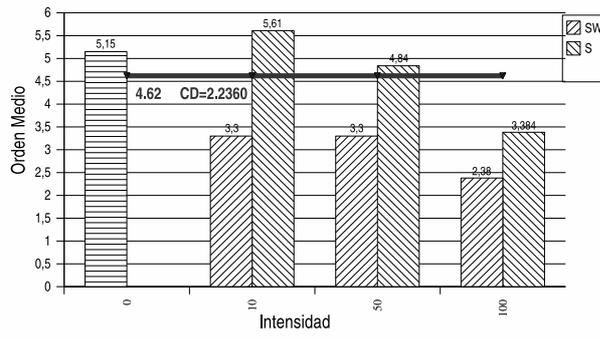
Cruce	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
ARIT	5.322758	2.23	R
BLX-0.3	2.478327	2.23	R
BLX-0.5	2.833856	2.23	R

Tabla 2.10: Test de Iman-Davenport con BD (SW y S)

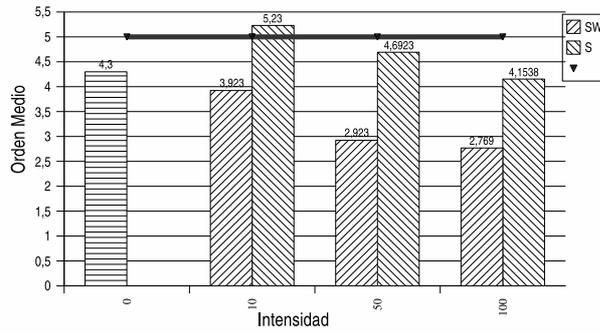
En primer lugar, vamos a aplicar el test Iman-Davenport para cada uno de los grupos. La Tabla 2.10 muestra los resultados. De dicha tabla se observa que para todos los métodos existe una diferencia significativa. Lo cual justifica un mayor estudio.

La Figura 2.20 muestra el resultado de aplicar el test de Bonferroni-Dunn. De dicha figura podemos observar que en todos los casos los resultados de aplicar el método Solis Wets son superiores a los resultados de aplicar el método Simplex.

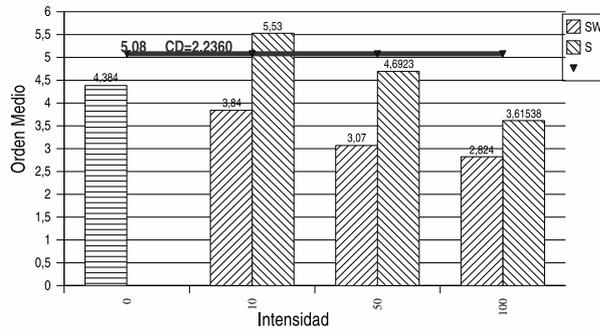
**Conclusiones** El mayor uso de la BL hace mejorar siempre los resultados con el Solis Wets, aunque para el Simplex es necesario un valor mínimo de intensidad. Si comparamos con el CHC, la BD presenta un mejor comportamiento que éste al incorporar/potenciar la BL, posiblemente debido a su diseño, que es capaz de introducir más eficientemente la mejora de la BL. Al igual que bajo CHC, el operador aritmético presenta malos resultados, por lo que no es recomendable su uso. Respecto al método de BL, el Solis Wets es el que ofrece mejores resultados, y, ya que en la BD aumentar la intensidad mejora los resultados, las mejores



(a) Cruce Aritmético



(b) Cruce BLX-0.3



(c) Cruce BLX-0.5

Figura 2.20: Test de Bonferroni-Dunn con Búsqueda Dispersa (SW y S)

variantes son aquellas con un alto valor de intensidad (*BD (BLX-0.3 SW 100)* y *BD (BLX-0.5 SW 100)*).

### 2.3.2.3. Estudio Experimental del CLPSO

En este apartado analizaremos el comportamiento de CLPSO y su evolución conforme se le incrementa la intensidad de la BL (aumentando así la importancia de la BL en la búsqueda). Para ello hemos implementado varias instancias del CLPSO que se distinguen del método de BL y la intensidad de la BL empleada para dicho método ( $n_{int}$ ).

Cada variante con BL se identifica como *BL*  $n_{int}$ , en donde *BL* identifica al método de BL empleado (*SW* para Solis Wets y *S* para Simplex), y  $n_{int}$  es la intensidad de la BL (10, 50 ó 100).

Las características de los algoritmos comparados son:

- Una población de 30 individuos.
- $c_1$  y  $c_2$  valen 1.
- $w_k = \frac{(0,9-0,2) \cdot (maxGen-k)}{maxGen} + 0,2$ , donde *maxGen* es la generación máxima.
- $V_{max} = 0,25 \cdot (X_{max} - X_{min})$ .
- Los operadores de cruce aritmético y *BLX* -  $\alpha$  (con  $\alpha=0.3$  y  $0.5$ ).
- Los valores de intensidad  $n_{int} = 0, 10, 500, 100$ .
- Los métodos de BL Simplex y Solis Wets.

Los parámetros anteriores se han realizado siguiendo las recomendaciones de los autores [LQSB03].

Dado que en este algoritmo no existe operador de cruce el número de variantes es lo suficientemente reducido (7) para realizar un análisis conjunto de todos, comparando ambos tipos de BL.

En primer lugar, aplicamos el test de Iman-Davenport para determinar si existen diferencias significativas entre ellos. Dicho resultado, reflejado en la Tabla 2.11, muestra que no existe ninguna diferencia significativa, por lo que no es necesario un estudio más detallado.

Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
1.413	2.23	A

Tabla 2.11: Test de Iman-Davenport con CLPSO

En conclusión, dado que la incorporación de la BL no implica una mejora significativa, no parece un AE prometedor con el método de hibridación empleado. Sin embargo, es justo comentar que:

- El método de hibridación seguido (el clásico de Hart) no es adecuado con este modelo. Existen modelos meméticos que hacen uso del modelo PSO (como [LS05]) que aplican la BL sólo al mejor, ya que en este modelo dicha mejora se propaga rápidamente al resto de la población.
- En problemas reales, la incorporación de una BL adaptada al problema sí puede suponer una mejora significativa.

#### 2.3.2.4. Estudio Experimental del Algoritmo Genético Generacional

En este apartado analizamos el comportamiento del AGG, prestando especial atención a la influencia del operador de selección, los distintos operadores de cruce, y su evolución conforme se le incrementa la intensidad de la BL (aumentando así la importancia de la BL en la búsqueda).

Hemos implementado varias instancias del AGG que se distinguen por el método de selección de los padres, el cruce, el método de BL y la intensidad de la BL empleada para dicho método ( $n_{int}$ ).

Las características de los algoritmos considerados son:

- Tamaño de la población  $Popsiz$  = 60.
- Los operador de selección de padres (TS, LR, RS y NAM).
- Los operadores de cruce aritmético y  $BLX - \alpha$ .
- Los métodos de BL Simplex y Solis Wets.
- Los operadores de mutación *No Uniforme* y *BGA*.

Debido al gran número de variantes se va a realizar un análisis por partes, ya que un estudio conjunto sería demasiado complejo para poder sacar conclusiones fácilmente.

Por tanto, este apartado se estructura de la siguiente forma:

1. Primero, realizaremos un análisis con los distintos operadores de mutación considerados, para identificar aquél que consideremos más adecuado.
2. A continuación, procedemos a estudiar para cada BL y cada uno de los valores de  $n_{int}$  las distintas combinaciones de selección y cruce, identificando en cada caso la mejor combinación.
3. Una vez determinado para cada BL e intensidad las mejores combinaciones de selección y cruce, se eligen las mejores combinaciones para cada  $n_{int}$ , y se comparan entre sí (para los distintos métodos de BL).

**Elección del operador de mutación** Aunque en este estudio hemos intentado comparar los distintos operadores de forma conjunta (para poder considerar posibles sinergias entre ellos), la elección del operador de mutación se hará de forma independiente. Lo hacemos así por dos motivos: Por un lado, para reducir la complejidad del estudio comparativo; y por el otro, porque al incorporar diversidad de forma totalmente aleatoria la posibilidad de que exista una sinergia con otros operadores es reducida.

La comparación se ha aplicado con el modelo tradicional del AGG, usando el método de selección *Ranking Lineal* o LR. Para cada valor de  $n_{int}$  y operador de cruce se han comparado los resultados aplicando cada uno de los dos operadores.

La Tabla 2.12 muestra para cada intensidad de BL los errores acumulados obtenidos para cada uno de los operadores de mutación (marcándose en negrita aquel con mejor valor), y la diferencia obtenida.

De dicha tabla puede observarse que:

- En la práctica totalidad de los casos, los resultados aplicando la mutación BGA son mejores que aplicando la mutación no uniforme con lo que la mutación BGA es más conveniente al ofrecer menor error acumulado.
- La mejora obtenida por la mutación BGA es mayor conforme aumenta la diversidad del cruce, lo que confirma que ambos tipos de diversidad son

complementarios. Se debe a su diferente naturaleza: mientras que la del cruce está orientada hacia ciertas zonas (determinado por los padres elegidos), la diversidad de la mutación es totalmente aleatoria.

Cruce	No Uniforme	BGA	Diferencia	Cruce	No Uniforme	BGA	Diferencia
Sin BL				SW 10			
ARIT	9.18	<b>6.7</b>	2.48	ARIT	<b>2.04</b>	2.53	-0.49
BLX-0.3	5.60	<b>4.85</b>	0.75	BLX-0.3	8.63	<b>2.24</b>	6.39
BLX-0.5	7.00	<b>4.24</b>	2.77	BLX-0.5	12.95	<b>2.23</b>	10.72
SW 50				SW 100			
ARIT	7.38	<b>5.84</b>	1.54	ARIT	7.00	<b>5.85</b>	1.15
BLX-0.3	8.45	<b>6.62</b>	1.82	BLX-0.3	9.27	<b>6.8</b>	2.46
BLX-0.5	10.86	<b>7.26</b>	3.60	BLX-0.5	11.09	<b>7.51</b>	3.57

Tabla 2.12: Diferencia de EAc entre las variantes con la mutación No Uniforme y la BGA para AGG con SW

Por tanto, vistos los resultados obtenidos, seleccionaremos la mutación BGA por su mejor comportamiento.

**Análisis del operador cruce y el método de selección** En este apartado se realizará un estudio para los distintos valores de intensidad de la BL ( $n_{int}$ ), de los distintos métodos de selección y de cruce.

Las variantes se identifican como “*Sel-Cruce BL  $n_{int}$* ”. *Sel* toma los valores NAM, LR, RS y TS (de menor a mayor presión selectiva). *Cruce* toma valores ARIT (Aritmético), BLX-0.3 y BLX-0.5; *BL* toma como valores SW (Solis Wets) y S (Simplex); y  $n_{int}$  toma los valores 10, 50 y 100. Denotamos “*Sel-Cruce*” a las variantes que no hacen uso del método de mejora. En total, son 84 variantes (12 por cada valor de intensidad). El alto número de variantes hace conveniente un análisis para cada valor de  $n_{int}$ , de 12 variantes en cada uno.

En primer lugar, aplicamos el test de Iman-Davenport para determinar si existen diferencias significativas en cada grupo. Dicho resultado, reflejado en la Tabla 2.13, muestra que existe entre cada grupo diferencias significativas, por lo que es necesario un estudio más detallado para cada valor de BL y  $n_{int}$ .

BL	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
Sin BL	7.65596	1.86	R
SW 10	10.07368	1.86	R
SW 50	7.67196	1.86	R
SW 100	7.87146	1.86	R
S 10	11.06115	1.86	R
S 50	11.88303	1.86	R
S 100	9.25739	1.86	R

Tabla 2.13: Test de Iman-Davenport con AGG y BGA

**AGG con Solis Wets** En este apartado utilizaremos dos tests diferentes:

- El test de Holm, para comparar las variantes con la misma intensidad, ya que en este caso sólo nos interesa una ordenación de las variantes comparadas con aquella con mejor orden medio, y este test es más potente que el test de Bonferroni-Dunn. Además, el test de Holm ha identificado que la mejor combinación es la misma para todos los valores de intensidad (*TS BLX-0.5*, facilitando la comparación).
- El test de Bonferroni-Dunn para comparar entre sí la influencia de la BL para el conjunto de variantes que hayan demostrado no ser estadísticamente peores que la mejor (*TS BLX-0.5*). Del total de 12 variantes se han escogido las 4 que ofrecían mejores resultados según el test de Holm anterior. Para mostrar la influencia de la intensidad de la búsqueda local, se ha establecido una gráfica mostrando la evolución de los resultados para cada  $n_{int}$ .

Las Tablas 2.14 y 2.15 muestran los resultados de aplicar el test de Holm para cada una de las intensidades. De dichas tablas hemos extraído las siguientes conclusiones:

- *Influencia del operador de cruce.* A diferencia de los anteriores, no existe un predominio claro de los operadores de cruce. Sea cual sea el cruce, sino se elige un adecuado método de selección, no se obtendrán buenos resultados.
- *Influencia del método de selección.* Las diferencias entre los distintos métodos de selección se muestran mucho mayores que entre distintos métodos de

i	Algoritmo	z	p	$\alpha/i$	Aceptada o Rechazada
Sin BL					
11	NAM-BLX-0.5	5.222	1.77e-07	0.0045	R
10	NAM-ARIT	3.807	0.000140	0.0050	R
9	RS-ARIT	3.807	0.000140	0.0056	R
8	RS-BLX-0.5	3.590	0.000331	0.0063	R
7	NAM-BLX-0.3	3.372	0.000745	0.0071	R
6	RS-BLX-0.3	2.230	0.025740	0.0083	A
5	TS-ARIT	1.904	0.056943	0.0100	A
4	LR-ARIT	1.741	0.081759	0.0125	A
3	TS-BLX-0.3	1.197	0.231446	0.0167	A
2	LR-BLX-0.3	1.142	0.253350	0.0250	A
1	LR-BLX-0.5	0.381	0.703389	0.0500	A
SW 10					
11	NAM-BLX-0.5	5.494	3.93e-08	0.0045	R
10	RS-BLX-0.5	4.841	1.29e-06	0.0050	R
9	NAM-BLX-0.3	3.971	7.16e-05	0.0056	R
8	NAM-ARIT	3.209	0.001331	0.0063	R
7	RS-ARIT	3.209	0.001331	0.0071	R
6	RS-BLX-0.3	2.720	0.006535	0.0083	R
5	LR-ARIT	1.741	0.081759	0.0100	A
4	TS-ARIT	1.577	0.114705	0.0125	A
3	LR-BLX-0.5	1.251	0.210922	0.0167	A
2	LR-BLX-0.3	1.197	0.231446	0.0250	A
1	TS-BLX-0.3	0.490	0.624463	0.0500	A

Tabla 2.14: Test de Holm de AGG Versus mejor(TS-BLX-0.5) con Solis Wets, sin BL y SW 10

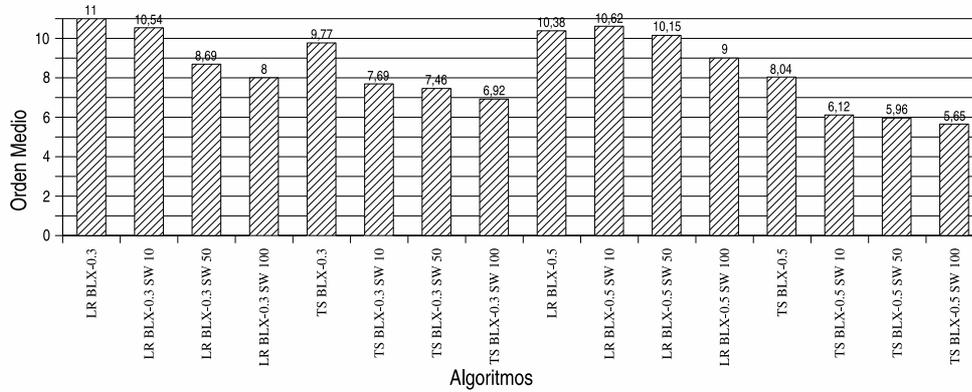
cruce, y se observa que sin BL es conveniente el uso de métodos de selección que (como LR o TS) incorporen una cierta presión selectiva.

Para valores bajos de  $n_{int}$  es incluso más decisivo: sin BL y con  $n_{int} = 10$ , los algoritmos estadísticamente peores son los que usan el NAM o el RS. Con  $n_{int} = 50, 100$  no existe un criterio tan claro, pero se observa que las peores son *NAM-BLX-0.3/5* y *RS-BLX-0.5*.

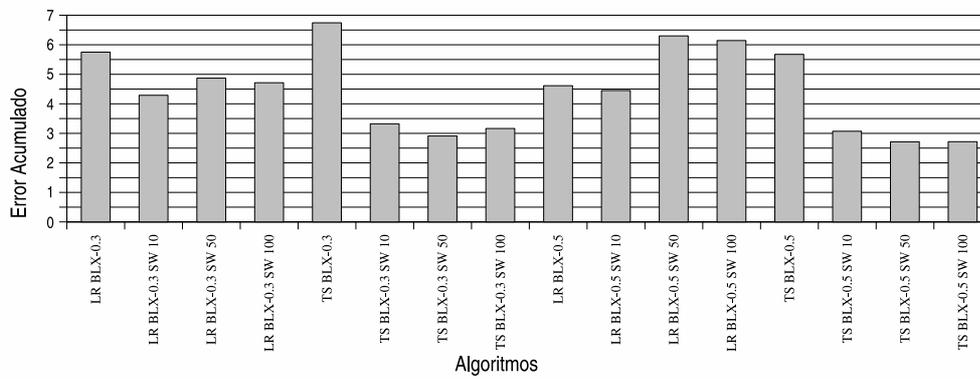
i	Algoritmo	z	p	$\alpha/i$	Acceptada o Rechazada
SW 50					
11	NAM-BLX-0.5	4.623	3.77e-06	0.0045	R
10	RS-BLX-0.5	3.971	7.16e-05	0.0050	R
9	NAM-BLX-0.3	3.644	0.000268	0.0056	R
8	NAM-ARIT	2.013	0.044164	0.0063	A
7	RS-BLX-0.3	1.958	0.050213	0.0071	A
6	RS-ARIT	1.686	0.091762	0.0083	A
5	LR-BLX-0.5	1.033	0.301387	0.0100	A
4	TS-ARIT	0.816	0.414562	0.0125	A
3	LR-ARIT	0.326	0.744154	0.0167	A
2	LR-BLX-0.3	0.272	0.785650	0.0250	A
1	TS-BLX-0.3	0.218	0.827763	0.0500	A
SW 100					
11	NAM-BLX-0.5	5.222	1.77e-07	0.0045	R
10	RS-BLX-0.5	4.678	2.90e-06	0.0050	R
9	NAM-BLX-0.3	4.351	1.35e-05	0.0056	R
8	NAM-ARIT	2.502	0.012347	0.0063	A
7	RS-BLX-0.3	2.448	0.014378	0.0071	A
6	RS-ARIT	2.339	0.019341	0.0083	A
5	LR-BLX-0.5	1.741	0.081759	0.0100	A
4	TS-ARIT	1.577	0.114705	0.0125	A
3	LR-BLX-0.3	1.251	0.210922	0.0167	A
2	LR-ARIT	1.197	0.231446	0.0250	A
1	TS-BLX-0.3	1.088	0.276658	0.0500	A

Tabla 2.15: Test de Holm de AGG Versus mejor(TS-BLX-0.5) con Solis Wets, con SW 50 y 100

La Figura 2.21 muestra el resultado de comparar las mejores combinaciones de selección y cruce. De dicha tabla puede observarse:



(a) Orden Medio



(b) Errores Acumulados

Figura 2.21: Comparación de las Mejores variantes del AGG con Solis Wets

- *Influencia de la BL.* Conforme se incrementa la BL, se obtienen mejores resultados en orden medio. Sin embargo, el aumentar la intensidad no implica una mejora en el EAc, excepto en la considerada mejor variante, *TS-BLX-0.5*. Los métodos de selección que presentan mejores resultados lo hacen para los distintos valores de intensidad. La influencia del cruce disminuye

conforme aumenta la BL (debido a la disminución de generaciones producidas al aumentar la intensidad  $n_{int}$ ).

- *Mejores variantes.* La mejor variante es clara, *TS-BLX-0.5*. A la hora de identificar la siguiente mejor variante, se observan dos tendencias. El TS y, posteriormente, el LR ofrecen los mejores resultados. además, para estos operadores el operador BLX-0.5 tiende a ofrecer mejores resultados que el BLX-0.3. Cuando se usan valores pequeños de intensidad (Sin BL) predomina la elección del cruce (por lo que las mejores son *LR-BLX-0.5* y *TS-BLX-0.5*), con más BL ( $n_{int} = 50$ ) no hay un predominio claro, mientras que con mayor BL ( $n_{int} = 100$ ) predomina el operador de selección (*TS-BLX-0.3* y *TS-BLX-0.5*). Elegiremos como mejores variantes *LR-BLX-0.5* y *TS-BLX-0.5*.

En conclusión, se observa que existe una sinergia entre los métodos de selección y el cruce, aunque la influencia del operador de selección es mayor que la del cruce. Debido a que el esquema del AGG mantiene cierta diversidad (por reemplazar toda la población en cada generación), es contraproducente que tanto el método de selección como el cruce fomenten la diversidad. Los métodos de selección con alta presión selectiva, como el TS o el LR, que orientan la búsqueda hacia las zonas más prometedoras encontradas, ofrecen los mejores resultados. El uso de un cruce como el *BLX-0.5* que aporta diversidad también contribuye a obtener buenos resultados.

**AGG con Simplex** Para mostrar la influencia de la intensidad de la BL, se ha establecido una gráfica mostrando la evolución de los resultados para cada  $n_{int}$ .

Las Tablas 2.16 y 2.17 muestran el resultado de Holm para cada una de las intensidades. De dichas tablas hemos extraído las siguientes conclusiones:

- *Influencia del operador de cruce.* Se observa que con operadores de selección con diversidad (NAM y RS) funcionan mejor un cruce con ligera diversidad, mientras que con el resto se obtienen mejores resultados con mayor diversidad (BLX-0.5). Sin embargo, se observa que el operador de cruce posee menor influencia que empleando el método Solis Wets.
- *Influencia del Método de Selección.* Con el método Simplex se observa que el operador de selección posee una gran influencia en la bondad del algoritmo.

i	Algoritmo	z	p	$\alpha/i$	Aceptada o Rechazada
Sin BL					
11	NAM-BLX-0.5	5.222	1.77e-07	0.0045	R
10	NAM-ARIT	3.807	0.000140	0.0050	R
9	RS-ARIT	3.807	0.000140	0.0056	R
8	RS-BLX-0.5	3.590	0.000331	0.0063	R
7	NAM-BLX-0.3	3.372	0.000745	0.0071	R
6	RS-BLX-0.3	2.230	0.025740	0.0083	A
5	TS-ARIT	1.904	0.056943	0.0100	A
4	LR-ARIT	1.741	0.081759	0.0125	A
3	TS-BLX-0.3	1.197	0.231446	0.0167	A
2	LR-BLX-0.3	1.142	0.253350	0.0250	A
1	LR-BLX-0.5	0.381	0.703389	0.0500	A
S 10					
11	NAM-BLX-0.5	5.439	5.35e-08	0.0045	R
10	RS-ARIT	5.167	2.37e-07	0.0050	R
9	NAM-BLX-0.3	4.297	1.73e-05	0.0056	R
8	NAM-ARIT	4.243	2.21e-05	0.0063	R
7	RS-BLX-0.5	3.862	0.000113	0.0071	R
6	RS-BLX-0.3	2.828	0.004678	0.0083	R
5	TS-ARIT	2.556	0.010574	0.0100	A
4	LR-ARIT	2.448	0.014378	0.0125	A
3	LR-BLX-0.3	1.305	0.191747	0.0167	A
2	TS-BLX-0.3	1.088	0.276658	0.0250	A
1	LR-BLX-0.5	1.033	0.301387	0.0500	A

Tabla 2.16: Test de Holm de AGG Versus mejor(TS-BLX-0.5) con Simplex, sin BL y con S 10

Todas las variantes que emplean las selecciones RS y NAM son estadísticamente peores que el mejor, por lo que se hace necesario el empleo de una selección con alta presión selectiva (como la TS o el LR).

La Figura 2.22 muestra el resultado de comparar las mejores combinaciones de selección y cruce. De dicha tabla puede observarse que conforme se incrementa la BL, mejores resultados se obtienen en orden medio. Sin embargo, el aumentar la intensidad no implica una mejora en el EAc, excepto en la considerada mejor variante, *TS-BLX-0.5*.

- *Influencia de la BL.* Se observa que al introducir la BL, los resultados empeoran. Sin embargo, a partir de un cierto  $n_{int}$ , los resultados empiezan a

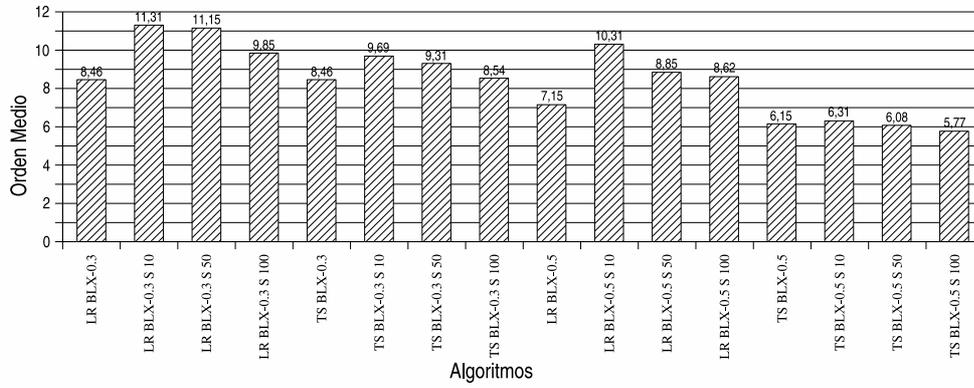
i	Algoritmo	z	p	$\alpha/i$	Aceptada o Rechazada
S 50					
11	RS-ARIT	5.276	1.32e-09	0.0045	R
10	NAM-BLX-0.5	4.841	1.29e-06	0.0050	R
9	NAM-ARIT	4.297	1.73e-05	0.0056	R
8	NAM-BLX-0.3	3.753	0.000175	0.0063	R
7	RS-BLX-0.5	3.536	0.000407	0.0071	R
6	RS-BLX-0.3	3.100	0.001933	0.0083	R
5	TS-ARIT	2.665	0.007693	0.0100	R
4	LR-ARIT	2.067	0.038741	0.0125	A
3	LR-BLX-0.3	0.925	0.355133	0.0167	A
2	TS-BLX-0.3	0.925	0.355133	0.0250	A
1	LR-BLX-0.5	0.272	0.785650	0.0500	A
S 100					
11	RS-ARIT	5.276	1.32e-07	0.0045	R
10	NAM-BLX-0.5	4.841	1.29e-06	0.0050	R
9	NAM-ARIT	4.297	1.73e-05	0.0056	R
8	NAM-BLX-0.3	3.753	0.000175	0.0063	R
7	RS-BLX-0.5	3.536	0.000407	0.0071	R
6	RS-BLX-0.3	3.100	0.001933	0.0083	R
5	TS-ARIT	2.665	0.007693	0.0100	R
4	LR-ARIT	2.067	0.038741	0.0125	A
3	LR-BLX-0.3	0.925	0.355133	0.0167	A
2	TS-BLX-0.3	0.925	0.355133	0.0250	A
1	LR-BLX-0.5	0.272	0.785650	0.0500	A

Tabla 2.17: Test de Holm de AGG Versus mejor(TS-BLX-0.5) con Simplex, con S 50 y S 100

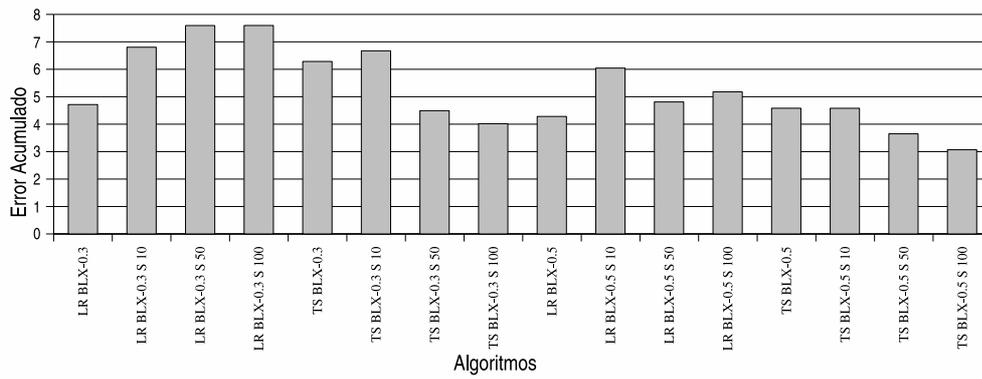
mejorar. Es decir, el método de BL Simplex requiere una cierta intensidad para empezar a producir una mejora en los resultados. Pero no se ha observado ninguna diferencia clara producida por el incremento de intensidad  $n_{int}$  entre los distintos métodos de selección.

- *Mejores variantes.* Las mejores variantes son, claramente, el *TS-BLX-0.5* y luego el *LR-BLX-0.5*.

**Elección de la Búsqueda Local (SW Versus S)** Al igual que con CHC y BD, vamos a comparar entre sí las variantes meméticas del AGG para cada



(a) Orden Medio



(b) Errores Acumulados

Figura 2.22: Comparación de las mejores variantes del AGG con Simplex

método de BL considerado. Sin embargo, dado el amplio número de variantes, vamos a comparar únicamente las mejores variantes (BLX-0.5 LR y BLX-0.5 TS).

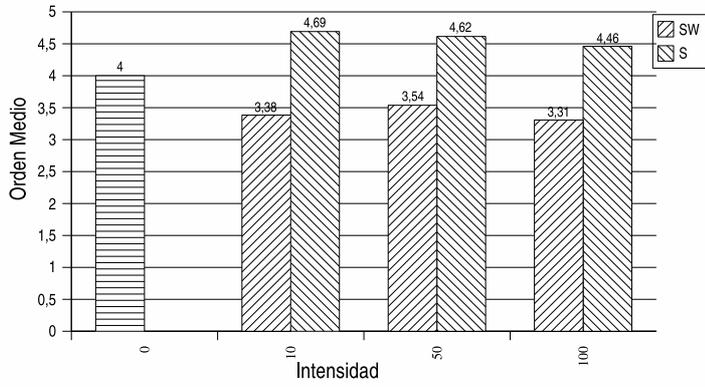
Selección	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
TS	0.994050	2.23	A
LR	1.174942	2.23	A

Tabla 2.18: Test de Iman-Davenport con AGG (SW y S)

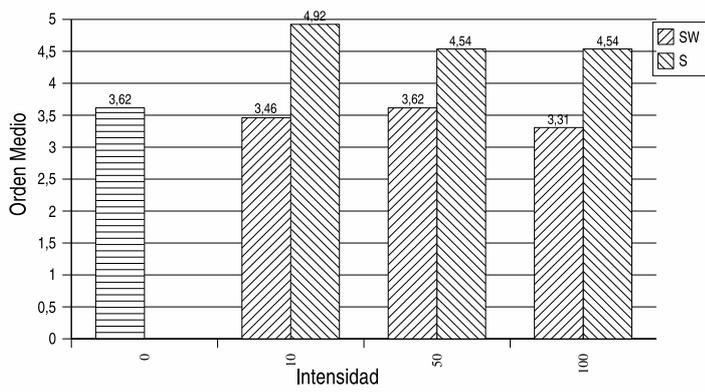
Vamos a aplicar el test de Iman-Davenport para cada uno de los grupos. La Tabla 2.18 muestra los resultados. De dicha tabla no se observa una diferencia significativa. Aun así, procedemos a comparar los órdenes medios para estudiar las diferencias existentes.

La Figura 2.23 muestra las diferencias en orden medio entre los dos operadores de BL. Se observa que, en la totalidad de los casos, el método de BL Solis Wets ofrece mejores resultados que el Simplex.

**Conclusiones** El operador que más determina el comportamiento general es el operador de selección, aunque existe una dependencia entre ambos factores. Las mejores combinaciones son las que combinan un operador de selección con alta presión selectiva (como la TS o el LR) junto con un operador de cruce que fomente la diversidad (como el BLX-0.5). Estos métodos de selección favorecen que los individuos mejorados por la BL sean utilizados para constituir la generación siguiente. Debido al criterio de reemplazo de los AGGs, este es un aspecto muy importante, ya que bajo otros operadores de selección la mejora obtenida por la BL no se introduciría en la población. Respecto a la BL, es el método Solis Wets el que obtiene los mejores resultados.



(a) Selección por Torneo (TS)



(b) Ranking Lineal (LR)

Figura 2.23: Test de Bonferroni-Dunn sobre distintos AGGs (SW y S)

### 2.3.2.5. Estudio Experimental del Algoritmo Genético Estacionario

Hemos implementado varias instancias del AGE que se distinguen por el método de cruce, el método de BL, los mecanismos de selección de padres y estrategia de reemplazo, y la intensidad de la BL empleada para dicho método ( $n_{int}$ ).

El problema comentado a la hora de identificar un adecuado AGG (excesivo número de variantes) se incrementa todavía más con respecto a los AGEs. En el caso de los AGEs los distintos operadores posibles (comentados en el apartado 2.2.5) serían:

- *Operadores de Cruce*: Aritmético, BLX-0.3 y BLX-0.5.
- *Operadores de Mutación*: BGA, No Uniforme.
- *Métodos de Selección de Padres*: NAM, RS, LR y TS.
- *Estrategias de Reemplazo*: DC, RTS, RW y WAMS.
- *Métodos de BL*: Solis Wets y Simplex.

Si lo consideramos todo da un total de 192 variantes (672 si se consideran las distintas intensidades de la BL). Como es un número excesivo, hemos aplicado una reducción de las variantes:

- Elección de la BL Solis Wets. Visto que los resultados del Solis han sido siempre mejores que el Simplex en los esquemas CHC, BD y AGG, hemos procedido a utilizar únicamente el Solis Wets, con lo que se reduce a cerca de la mitad el número de variantes.
- Comparación inicial del operador de mutación. Dado que el número de algoritmos seguía siendo muy alto, realizamos, al igual que con el AGG, una serie de experimentos iniciales para identificar la mutación, reduciéndose el número de variantes a 48 combinaciones (192 considerando las distintas intensidades).

Los valores adecuados para el resto de parámetros libres (operadores de cruce, operadores de selección y estrategias de reemplazo) se determinarán al igual que en el resto de AEs considerados (CHC, BD, CLPSO y AGG).

Este apartado está estructurado de la siguiente forma: Primero, se identifica el operador de mutación. Luego, se analiza el comportamiento para los distintos operadores de cruce. Luego, se analizan las distintas combinaciones de selección, identificando para cada intensidad las combinaciones más adecuadas. Finalmente, se comparan las mejores combinaciones con las distintas profundidades de BL entre sí.

**Operadores de mutación** Como con el AGG, compararemos los resultados de los métodos de mutación BGA y *No Uniforme* para una combinación clásica de selección y reemplazo (el LR para selección y el RW para el reemplazo).

Cruce	No Uniforme	BGA	Diferencia	Cruce	No Uniforme	BGA	Diferencia
Sin BL				SW 10			
ARIT	<b>8.62</b>	9.36	-0.74	ARIT	10.03	<b>8.06</b>	1.97
BLX-0.3	6.41	<b>4.02</b>	2.39	BLX-0.3	7.5	<b>5.89</b>	1.61
BLX-0.5	<b>3.38</b>	4.29	-0.91	BLX-0.5	4.96	<b>4.78</b>	0.18
SW 50				SW 100			
ARIT	9.56	<b>8.67</b>	0.89	ARIT	<b>10.66</b>	10.92	-0.26
BLX-0.3	6.39	<b>5.78</b>	0.61	BLX-0.3	<b>5.91</b>	5.99	-0.08
BLX-0.5	6.41	<b>5.78</b>	0.63	BLX-0.5	<b>5.55</b>	5.69	-0.14

Tabla 2.19: Errores Acumulados Mutación No Uniforme Versus BGA, para AGE con SW

La Tabla 2.19 muestra el error de aplicar cada uno de los operadores de BL (se destaca en negrita los menores errores). No se observa una gran diferencia entre ambas, aunque depende en gran medida de la intensidad. Se observa que la mutación BGA posee mejor comportamiento con  $n_{int} = 10, 50$  (y con  $n_{int} = 100$  las diferencias son muy reducidas), por lo que será el operador de mutación elegido.

**Operadores de cruce** Dado que existen 16 combinaciones de selección y reemplazo, y 3 operadores de cruce, sería excesivo comparar las 48 combinaciones simultáneamente entre sí. Por tanto, se ha comparado para cada profundidad de BL las distintas variantes, agrupadas según su estrategia de reemplazo.

La Tabla 2.20, que muestra el resultado de aplicar el test de Iman-Davenport para los distintos cruces, confirmando que existe en todos los casos una importante diferencia significativa, por lo que aplicamos test *post-hoc*. Para cada estrategia

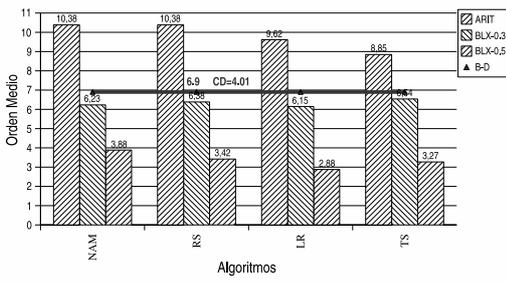
Estrategia de Reemplazo	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
Sin BL			
DC	17.799014	1.86	R
RTS	10.802642	1.86	R
RW	7.691326	1.86	R
WAMS	13.244081	1.86	R
SW 10			
DC	6.246131	1.86	R
RTS	3.701354	1.86	R
RW	6.388434	1.86	R
WAMS	5.305406	1.86	R
SW 50			
DC	4.152612	1.86	R
RTS	2.607566	1.86	R
RW	3.205747	1.86	R
WAMS	2.227739	1.86	R
SW 100			
DC	5.255980	1.86	R
RTS	4.939486	1.86	R
RW	6.808224	1.86	R
WAMS	2.828276	1.86	R

Tabla 2.20: Test de Iman-Davenport para AGE y Solis Wets, agrupados por su estrategia de reemplazo

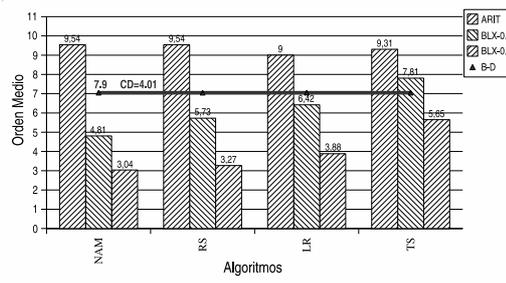
de reemplazo fijada, se identifican las variantes como “*Cruce Selección*”. Donde *Cruce* toma valores ARIT (Aritmético), BLX-0.3 y BLX-0.5; *Selección* toma como valores NAM, RS, LR y TS. Así pues, se comparan en grupos de 12 variantes.

Las Figuras 2.24, 2.25, 2.26 y 2.27 muestran el resultado de aplicar el test de Bonferroni-Dunn para los valores de intensidad  $n_{int} = 0, 10, 50$  y  $100$  respectivamente. De los resultados extraemos las siguientes conclusiones:

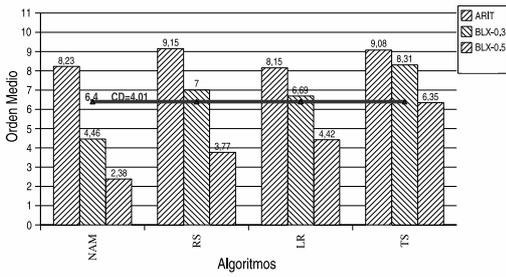
- El operador de cruce aritmético no presenta buenos resultados. La mayoría de las variantes que lo usan son estadísticamente peores.
- El cruce *BLX-0.3* presenta un comportamiento discreto. Para valor bajos de BL (sin BL o con  $n_{int} = 10, 50$ ) los resultados son claramente peores que los del *BLX-0.5*, mientras que con  $n_{int} = 100$  no hay diferencia clara entre ambas.



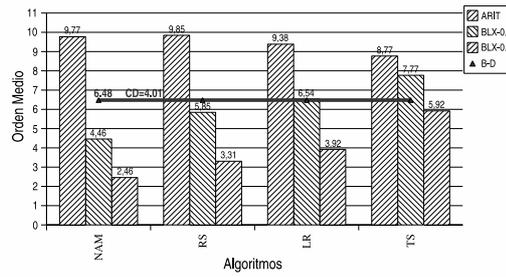
(a) Estrategia de Reemplazo DC



(b) Estrategia de Reemplazo RTS

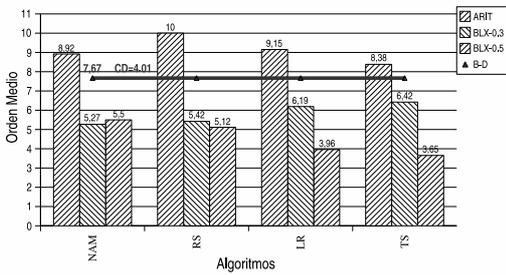


(c) Estrategia de Reemplazo RW

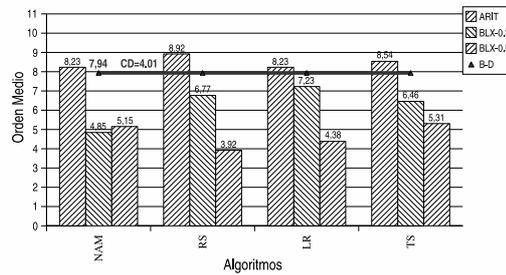


(d) Estrategia de Reemplazo WAMS

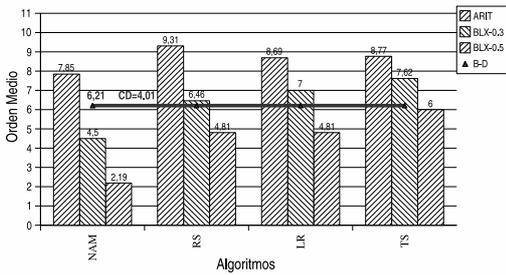
Figura 2.24: Test de Bonferroni-Dunn sobre el AGE Sin BL



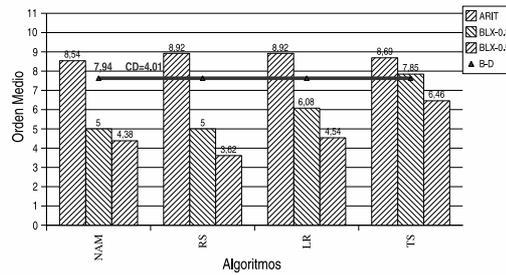
(a) Estrategia de Reemplazo DC



(b) Estrategia de Reemplazo RTS

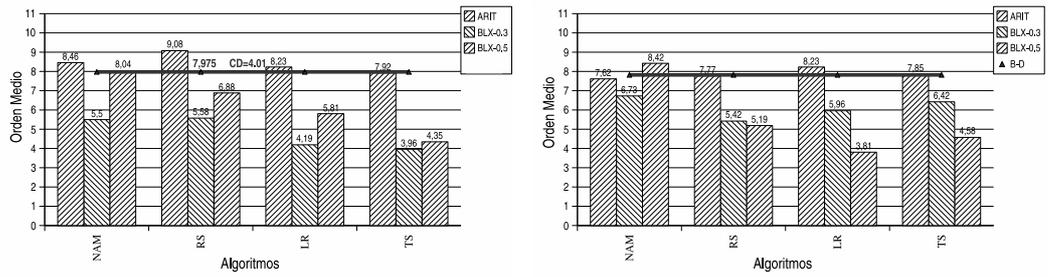


(c) Estrategia de Reemplazo RW



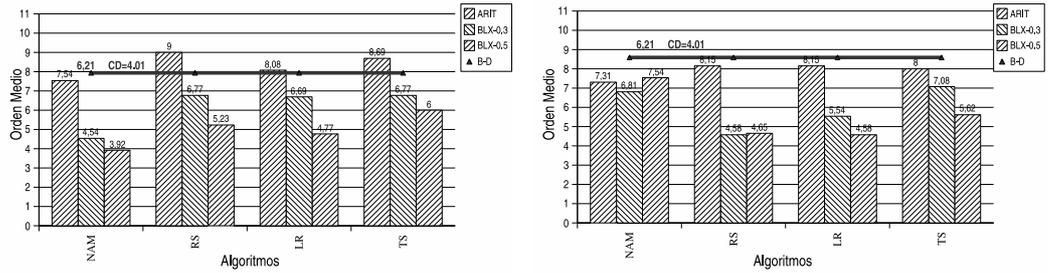
(d) Estrategia de Reemplazo WAMS

Figura 2.25: Test de Bonferroni-Dunn sobre el AGE SW 10



(a) Estrategia de Reemplazo DC

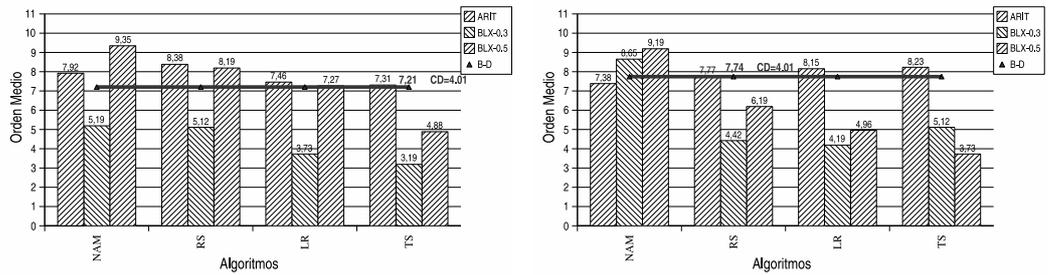
(b) Estrategia de Reemplazo RTS



(c) Estrategia de Reemplazo RW

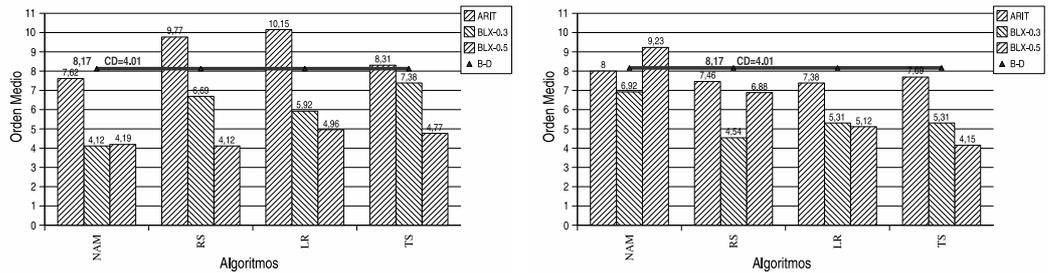
(d) Estrategia de Reemplazo WAMS

Figura 2.26: Test de Bonferroni-Dunn sobre el AGE SW 50



(a) Estrategia de Reemplazo DC

(b) Estrategia de Reemplazo RTS



(c) Estrategia de Reemplazo RW

(d) Estrategia de Reemplazo WAMS

Figura 2.27: Test de Bonferroni-Dunn sobre el AGE SW 100

- El cruce *BLX-0.5* presenta prácticamente siempre los mejores resultados, y lo emplea la variante con mejor orden medio (Excepto con  $n_{int}$  alta y con el reemplazo DC).

En conclusión, utilizaremos en lo sucesivo el cruce *BLX-0.5* ya que, aunque igual antes, para alguna estrategia de reemplazo puede ir mejor el *BLX-0.3*, el *BLX-0.5* presenta un comportamiento más estable.

**Análisis de las combinaciones selección de padres y reemplazo** Ya que hemos identificado los valores adecuados de mutación (BGA) y del cruce (BLX-0.5), procedemos a analizar el comportamiento para las distintas combinaciones de selección de padres y estrategia de reemplazo, y su evolución conforme se le incrementa la intensidad de la BL.

Las variantes se identifican como “*Sel Replace*”. Donde *Sel* toma los valores NAM, LR, RS y TS (de menor a mayor presión selectiva). *Replace* toma valores DC, RTS, RW, WAMS.

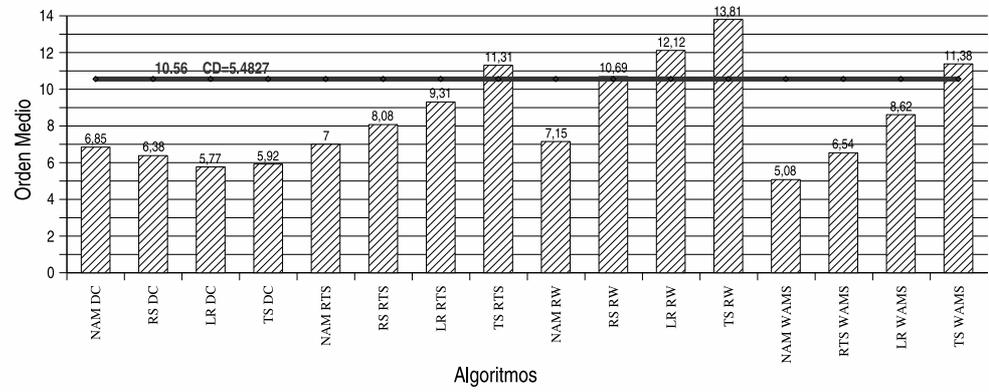
El alto número de variantes hace conveniente un análisis para cada valor de  $n_{int}$ , de 16 variantes en cada uno (resultantes de combinar las 4 posibilidades de selección y las 4 estrategias de reemplazo).

En primer lugar, aplicamos el test Iman-Davenport para determinar si existen diferencias significativas. La Tabla 2.21 muestra el resultado para cada intensidad, en donde se observa que efectivamente existen diferencias muy significativas entre las distintas combinaciones de selección y reemplazo, para cada una de las intensidades.

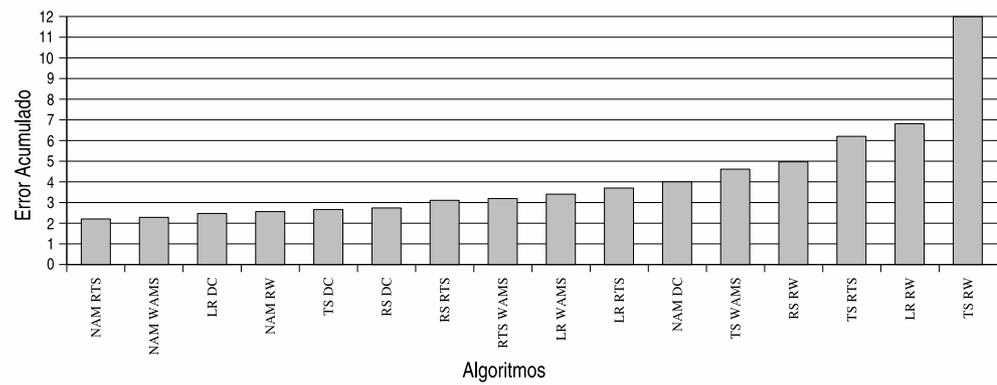
Búsqueda Local	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
Sin BL	5.290520	1.72	R
SW 10	2.863228	1.72	R
SW 50	2.948619	1.72	R
SW 100	7.671345	1.72	R

Tabla 2.21: Test de Iman-Davenport para AGE con BLX-0.5

**Análisis Selección-Reemplazo sin BL** La Figura 2.28 muestra el resultado de comparar las distintas variantes sin BL. De sus resultados concluimos que:



(a) Test de Bonferroni-Dunn



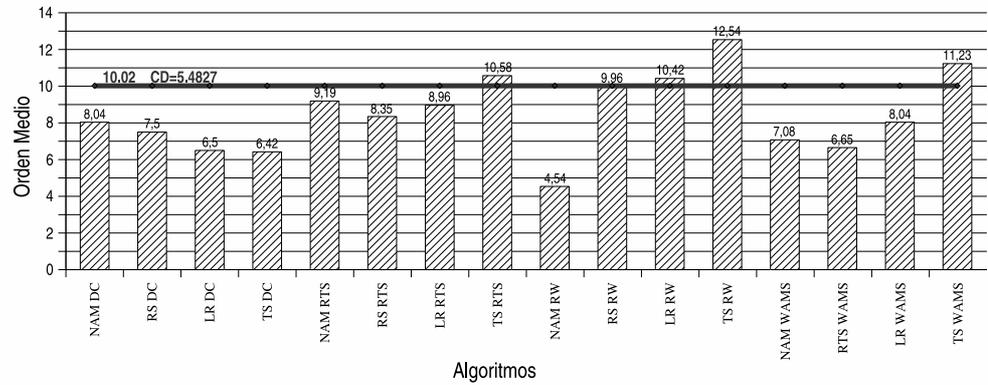
(b) Errores Acumulados

Figura 2.28: Comparación variantes del AGE con BLX-0.5, sin BL

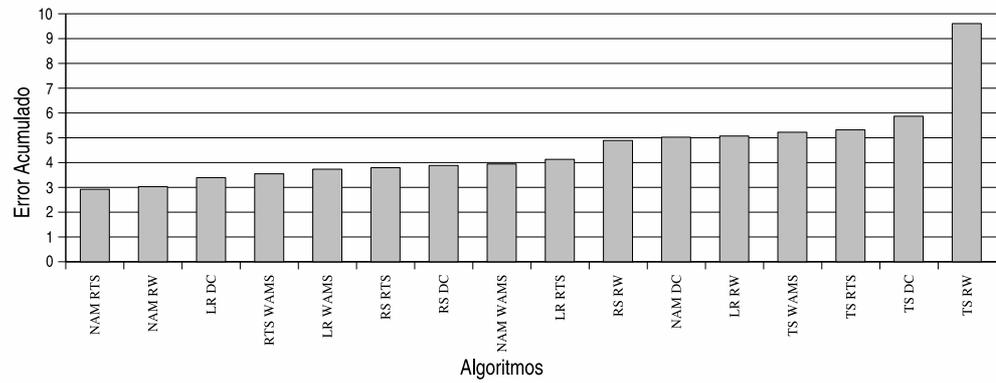
- *Alta influencia de la estrategia de reemplazo.* En general, la utilización de la estrategia de reemplazo DC ofrece buenos resultados, lo que muestra la conveniencia de utilizar estrategias de reemplazo que mantengan altos niveles de diversidad. Se obtienen los mejores resultados con DC y WAMS, aunque no en la totalidad de los casos, ya que depende siempre del operador de selección. En general, es DC y WAMS los que presentan los menores EAcs.
- *Aumento de la presión selectiva.* En la mayoría de estrategias, conforme mayor es la presión selectiva, peores resultados ofrece. Sin embargo, DC, por su criterio de sustituir al padre más parecido, permite mantener suficiente diversidad para contrarrestar el exceso de presión selectiva.
- *Mejores variantes.* Las mejores variantes en el caso de no aplicar la BL son *NAM-WAMS* y *LR-DC*.

**Análisis Selección-Reemplazo con SW 10** La Figura 2.29 muestra el resultado de comparar las distintas variantes con  $n_{int}=10$ . De sus resultados concluimos que:

- *Alta influencia de la estrategia de reemplazo.* La estrategia DC sigue ofreciendo buenos resultados, mostrándose RTS y RW con los peores resultados (aunque existe una combinación en particular, *NAM-RW* que ofrece los mejores resultados).
- *Aumento de la presión selectiva.* El comportamiento es dispar, para DC se obtienen mejores resultados aumentando la presión selectiva, mientras que para RW y WAMS conforme aumenta los resultados empeoran progresivamente.
- *Mejores variantes.* Se erige como la mejor combinación el uso combinado de una selección con diversidad y alta presión selectiva en el reemplazo (*NAM-RW*). La siguiente mejor es el *LR-DC*, que posee el tercer mejor orden medio y el tercer menor EAcs.

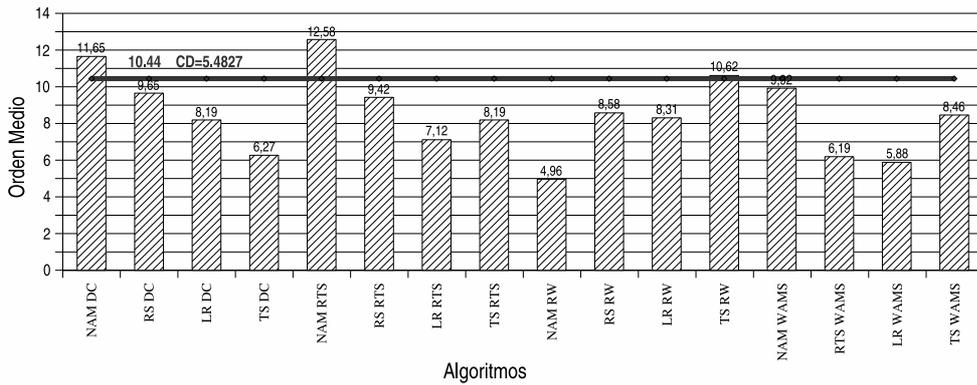


(a) Test de Bonferroni-Dunn

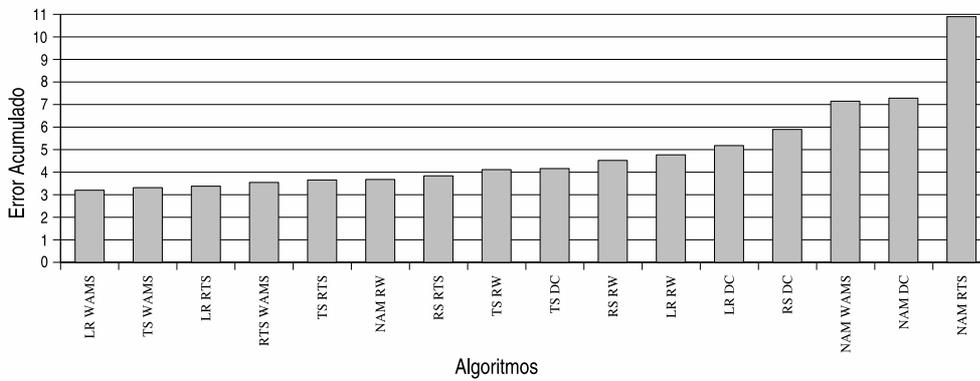


(b) Errores Acumulados

Figura 2.29: Comparación variantes del AGE con BLX-0.5, con Solis Wets,  $n_{int}=10$



(a) Test de Bonferroni-Dunn



(b) Errores Acumulados

Figura 2.30: Comparación variantes del AGE con BLX-0.5, con Solis Wets,  $n_{int} = 50$

**Análisis Selección-Reemplazo con SW 50** La Figura 2.30 muestra el resultado de comparar las distintas variantes con  $n_{int} = 50$ . De dicha gráfica podemos extraer las siguientes conclusiones:

- *Alta influencia de la estrategia de reemplazo:* Se obtienen los mejores resultados con DC y WAMS, aunque no en la totalidad de los casos, depende siempre del operador de selección. mejores EAc.

- *Aumento de la presión selectiva.* El comportamiento es dispar, para DC se obtienen mejores resultados aumentando la presión selectiva, mientras que para WAMS y RTS existe un nivel de equilibrio (mejoran hasta un punto en el que la presión es excesiva y empieza a empeorar de nuevo). Con el RW conforme aumenta la presión los resultados empeoran progresivamente.
- *Mejores variantes.* Determinamos como mejores variantes *NAM-RW* y *LR-WAMS* por poseer los menores órdenes medios, y bajo EAc.

**Análisis Selección-Reemplazo con SW 100** La Figura 2.31 muestra el resultado de comparar las distintas variantes con SW 100. De dicha gráfica podemos extraer las siguientes conclusiones:

- *Comportamiento de la estrategia de reemplazo.* En este caso, el uso de la estrategia RW ofrece los mejores resultados (para todos los operadores de selección). Este comportamiento es debido a que un alto valor de  $n_{int}$  implica un número muy bajo de iteraciones, por lo que es conveniente un método de reemplazo con alta presión selectiva como el RW para facilitar la convergencia. Debido igualmente al escaso número de iteraciones los resultados mejoran al usar una selección con alta presión selectiva.

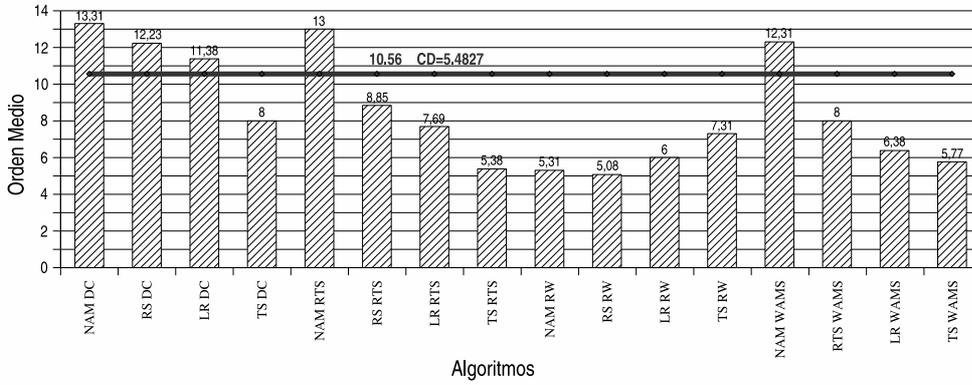
Respecto a los EAcs son las variantes con WAMS las que ofrecen los menores valores.

- *Mejores variantes.* Para los dos criterios seguidos (orden medio y EAc) las diferencias son muy altas, por lo que elegir un par de mejores variantes no es directo. Al final, se han elegido las variantes *TS-RTS* y *NAM-RW*.

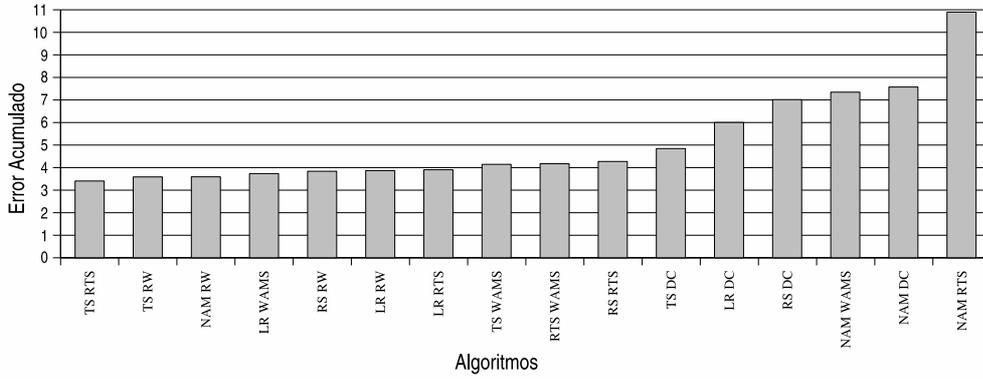
**Comparando las mejores variantes entre sí** Hasta ahora se han determinado para cada valor  $n_{int}$  el par de variantes que ofrecían los mejores resultados para dicho valor de intensidad. Eso permite identificar variantes prometedoras, pero no es suficiente a la hora de determinar una determinada combinación de selección y reemplazo.

Como conclusión de los estudios realizados para cada profundidad de BL obtenemos:

- El operador de reemplazo posee mayor influencia que el operador de selección. Sin embargo, existe una sinergia entre ambos factores: Una selección



(a) Test de Bonferroni-Dunn



(b) Errores Acumulados

Figura 2.31: Comparación variantes del AGE con BLX-0.5, con Solis Wets,  $n_{int}=100$

con mucha presión selectiva (como TS o LR) se beneficia de un mecanismo de reemplazo que mantenga diversidad (como DC), y una selección con poca presión (como NAM) se beneficia de un mecanismo de reemplazo con mayor presión (como RW).

- La bondad de las combinaciones depende del nivel de intensidad de la BL. Para una baja intensidad una estrategia de reemplazo como DC ofrece buenos resultados, mientras que con alta intensidad, las mejores utilizan estrategias de reemplazo con alta presión selectiva (RW, RTS).
- Mejores variantes. Para cada valor de intensidad se han elegido las dos variantes que ofrecían mejores resultados. La Tabla 2.22 muestra las elecciones. Se puede observar que existen variantes que son adecuadas para un rango de intensidad amplio (como el *NAM-RW*).

Profundidad	Mejor Variante	Segunda Mejor
Sin BL	NAM WAMS	LR DC
SW 10	NAM RW	TS/LR DC
SW 50	NAM RW	LR WAMS
SW 100	TS RTS	NAM RW

Tabla 2.22: Mejores Combinaciones de Selección-Reemplazo con el AGE

La Tabla 2.22 muestra para cada intensidad de BL la combinación que ha ofrecido los mejores resultados. Dentro de éstas, nos interesa conocer las más robustas, para lo que vamos a considerar las 5 variantes, y vamos a compararlas entre sí para cada valor de intensidad.

La Figura 2.32 muestra para cada valor de intensidad los órdenes medios. A pesar de calcularse de forma separada (por intensidad), los resultados son mostrados de forma conjunta, para mostrar si conforme se incrementa las variantes mejoran posiciones, se mantienen o empeoran.

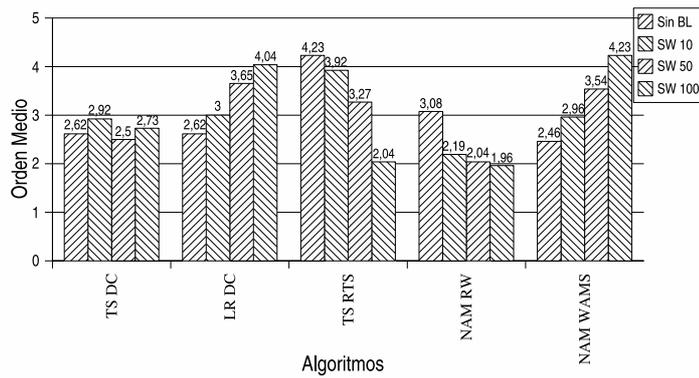


Figura 2.32: Orden Medio de las mejores AGEs con BLX-0.5 y SW, calculado para cada valor de intensidad

De dicha figura extraemos las siguientes conclusiones:

- Se observa que dos variantes, *LR-DC* y *NAM-WAMS* empeoran claramente sus posiciones conforme se incrementa la BL. Este comportamiento hace que las descartemos.
- Aunque las combinaciones *TS-RTS* y *LR-WAMS* presenta una tendencia descendente clara, los resultados para la valores iniciales de intensidad son demasiados altos, por lo que tampoco son adecuados.
- Las combinaciones *NAM-RW* y *TS-DC* son las que presentan un comportamiento más robusto manteniendo un buen orden medio, por lo que serán las elegidas.

**Conclusiones** Este apartado ha sido más largo que los anteriores debido al amplio número de variantes consideradas, pero el objetivo ha sido el mismo: observar la influencia de la BL e identificar las variantes más adecuadas. Tras un análisis en varias etapas, se han identificado como las mejores combinaciones el empleo de la BL *Solis Wets*, la mutación *BGA* y el operador de cruce *BLX-0.5*. Una vez identificado éstos, se han estudiado las distintas combinaciones de selección y reemplazo, y se han elegido, como las más adecuadas, las combinaciones *NAM-RW* y *TS-DC*, por sus buenos resultados.

### 2.3.3. Análisis Comparativo entre los distintos AEs

Tras identificar para cada modelo de AE las variantes más adecuadas, procedemos en este estudio a compararlas entre sí para analizar la conveniencia de cada uno de los AEs para su uso dentro de un AM.

Recapitulando, los AEs que van a ser comparados son:

- El algoritmo CHC con los operadores de cruce BLX-0.3 y BLX-0.5 (a los que llamaremos *CHC (BLX-0.3)* y *CHC (BLX-0.5)*, respectivamente).
- LBD con los operadores de cruce BLX-0.3 y BLX-0.5 (llamados *BD (BLX-0.3)* y *BD (BLX-0.5)*, respectivamente).
- El algoritmo CLPSO.
- Los AGGs con el operador de cruce BLX-0.5, mutación BGA con las selección de ranking lineal y torneo (llamados *AGG (LR-BLX-0.5)* y *AGG (TS-BLX-0.5)*, respectivamente).
- Los AGEs con el cruce BLX-0.5, mutación BGA y con las combinaciones de selección y reemplazo *NAM-RW* y *TS-DC* (llamados *AGE (NAM-RW-BLX-0.5)* y *AGE (TS-DC-BLX-0.5)*, respectivamente).

En total son 9 variantes, que aplicaremos con SW e intensidades  $n_{int}=0$  (sin BL), 10, 50 y 100. Las analizamos inicialmente de forma separada para cada intensidad, para mostrar cómo la relación entre ellas varía conforme aumenta la BL.

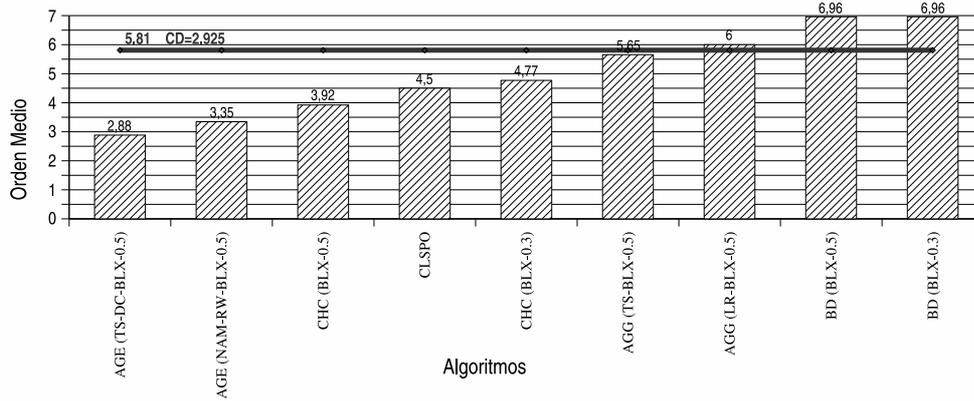
BL	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
Sin BL	5.059937	2.03	R
SW 10	6.612620	2.03	R
SW 50	3.283552	2.03	R
SW 100	3.749418	2.03	R

Tabla 2.23: Test de Iman-Davenport de las mejores propuestas

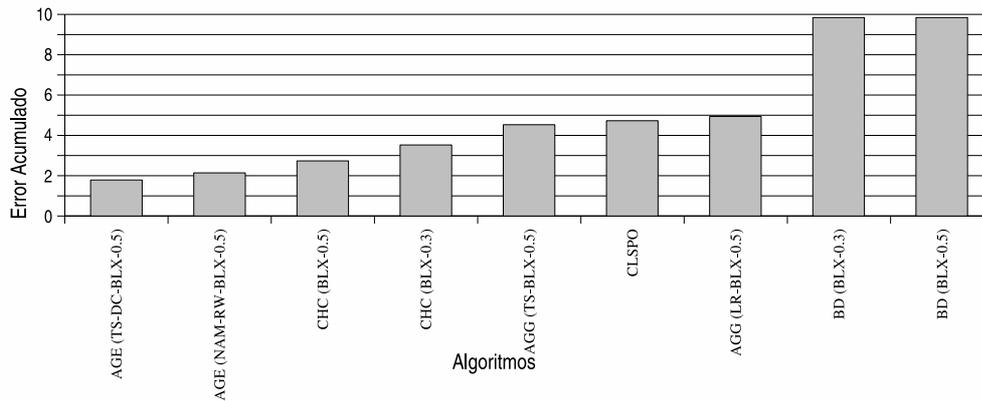
Como siempre, inicialmente aplicamos el test de Iman-Davenport para identificar si existe una diferencia significativa entre ellos.

En la Tabla 2.23 se observa que existe una diferencia significativa para cada intensidad. A continuación, estudiaremos el resultado para cada intensidad de BL (aplicando tanto el test de Bonferroni-Dunn como el test de Holm).

**AEs sin BL** En primer lugar vamos a analizarlos sin BL. Esta análisis podría ser de utilidad para su uso no memético, aunque en nuestro caso nos sirve principalmente para medir la evolución de los algoritmos.



(a) Test de Bonferroni-Dunn



(b) Errores Acumulados

Figura 2.33: Comparación de las mejores variantes de los AEs sin BL

La Tabla 2.33 muestra el resultado de comparar las dos mejores variantes de

i	Algoritmo	z	p	$\alpha/i$	Aceptada o Rechazada
8	BD (BLX-0.3)	3.795	0.000147	0.0062	R
7	BD (BLX-0.5)	3.867	0.000110	0.0071	R
6	AGG (LR-BLX-0.5)	2.900	0.003728	0.0083	R
5	AGG (TS-BLX-0.5)	2.578	0.009937	0.0100	R
4	CHC (BLX-0.3)	1.754	0.079348	0.0125	A
3	CLPSO	1.503	0.132622	0.0167	A
2	CHC (BLX-0.5)	0.966	0.333666	0.0250	A
1	AGE (NAM-RW.BLX-0.5)	0.429	0.667436	0.0500	A

Tabla 2.24: Test de Holm de los distintos AEs respecto al mejor: AGE (TS-DC-BLX-0.5), sin BL

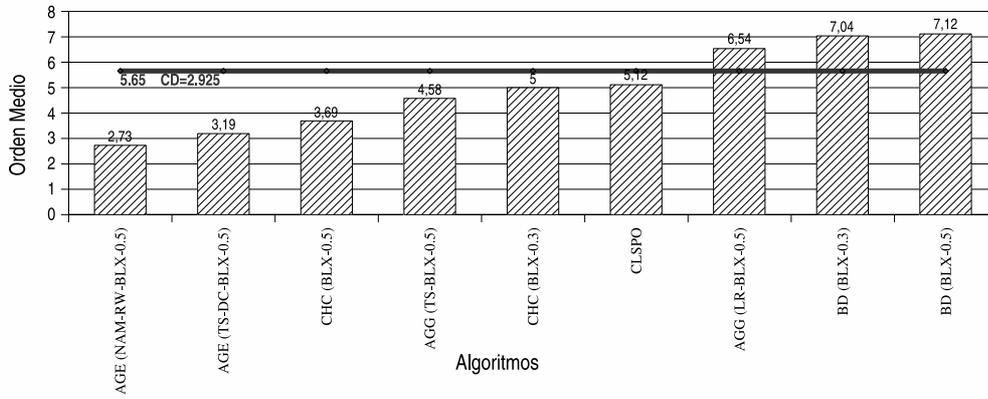
cada AE sin BL. De dicha tabla se puede observar que:

- No es recomendable el uso de la BD ni del AGG sin BL. En el caso de BD es previsible ya que es un algoritmo diseñado para su uso con BL, por lo que es esperable que sin BL se comporte peor que otros. En el AGG parece ser que los malos resultados derivan de su criterio de reemplazar todos los individuos de una población por los de la siguiente.
- Los mejores algoritmos son los que usan el enfoque del AGE y el CHC o CLPSO, sin diferencia significativa entre ellos. De entre estos, los mejores resultados son obtenidos por los AGEs, tanto en orden medio como en error acumulado. En particular, el mejor es el AGE (TS-DC-BLX-0.5).

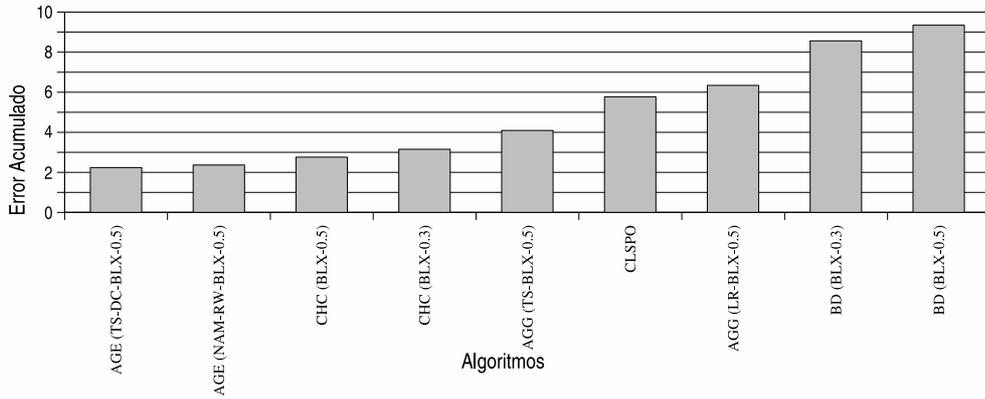
Para profundizar más en los resultados, hemos aplicado el test de Holm. La Tabla 2.24 muestra los resultados. El test de Holm no sólo confirma los resultados del test de Bonferroni-Dunn, sino que detecta una diferencia significativa entre AGE (TS-DC-BLX-0.5) y AGG (TS-BLX-0.5), diferencia no detectada por éste.

**SW 10** La Figura 2.34 muestra el resultado de comparar las dos mejores variantes de cada AE SW 10. De dicha figura se puede observar que:

- Los mejores resultados siguen siendo los obtenidos por el AGE, pero en esta ocasión el mejor es el AGE (NAM-RW-BLX-0.5).



(a) Test de Bonferroni-Dunn



(b) Errores Acumulados

Figura 2.34: Comparación de las mejores variantes de los AEs, con Solis Wets y  $n_{int}=10$

i	Algoritmo	z	p	$\alpha/i$	Aceptada o Rechazada
8	BD (BLX-0.3)	4.081	0.000044	0.0063	R
7	BD (BLX-0.5)	4.010	0.000060	0.0071	R
6	AGG (LR-BLX-0.5)	3.544	0.000392	0.0083	R
5	CLPSO	2.219	0.026421	0.0100	A
4	CHC (BLX-0.3)	2.112	0.034640	0.0125	A
3	AGG (TS-BLX-0.5)	1.719	0.085673	0.0167	A
2	CHC (BLX-0.5)	0.895	0.370710	0.0250	A
1	AGE (TS-DC-BLX-0.5)	0.429	0.667436	0.0500	A

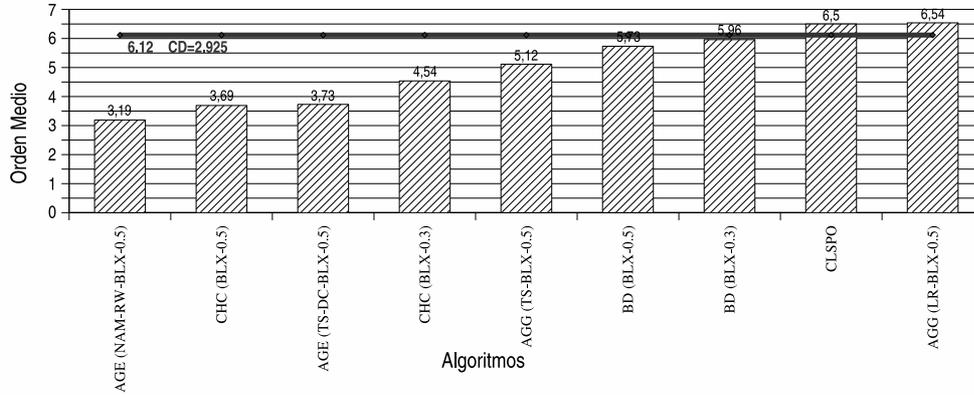
Tabla 2.25: Test de Holm de los distintos AEs respecto al mejor: AGE (NAM-RW-BLX-0.5) (SW 10)

- Siguen siendo los modelos AGG y BD estadísticamente peores, por los mismos motivos que sin BL (la intensidad es insuficiente para cambiar la tendencia).
- Los mejores algoritmos son los que usan AGE y CHC (y el AGG con TS), sin diferencia significativa entre ellos.

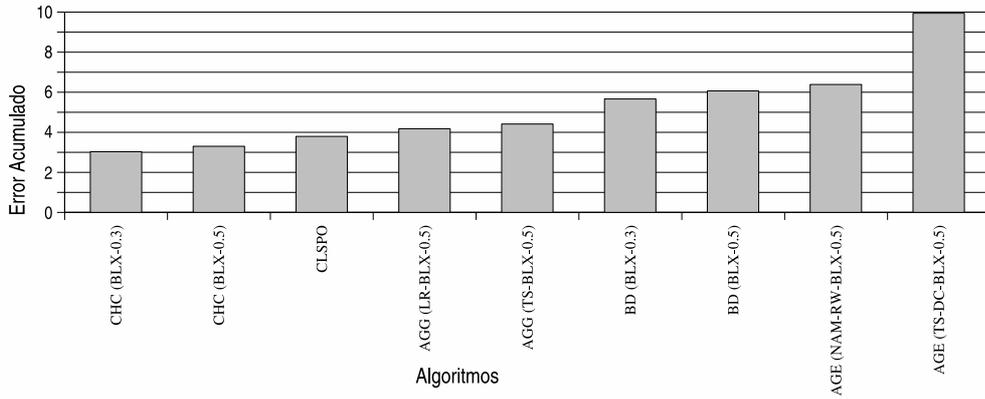
Hemos aplicado también el test de Holm. La Tabla 2.25 muestra los resultados, que confirma los resultados del test de Bonferroni-Dunn.

**SW 50** La Figura 2.35 muestra el resultado de comparar las dos mejores variantes de cada AE SW 50. De dicha figura se puede observar que:

- No existe una diferencia clara entre AGEs y los que usan CHC. Mientras que el *AGE (NAM RW BLX-0.5)* posee el menor orden medio, los que emplean CHC posee menor EAc.
- CLPSO sufre un empeoramiento relativo (posee el mayor EAc, y segundo peor orden medio). Conforme aumenta la intensidad, mientras que el resto de esquemas se obtiene una mejora, en CLPSO los resultados no mejoran de forma clara.
- La BD se comporta mejor conforme mayor es la intensidad de la BL.
- Los mejores algoritmos son los que usan el enfoque de AGE, CHC y BD, sin diferencia significativa entre ellos.



(a) Test de Bonferroni



(b) Errores Acumulados

Figura 2.35: Comparación de las mejores variantes de los AEs, con Solis Wets y  $n_{int}=50$

Hemos aplicado también el test de Holm. La Tabla 2.26 muestra los resultados, que confirma los resultados de Bonferroni-Dunn.

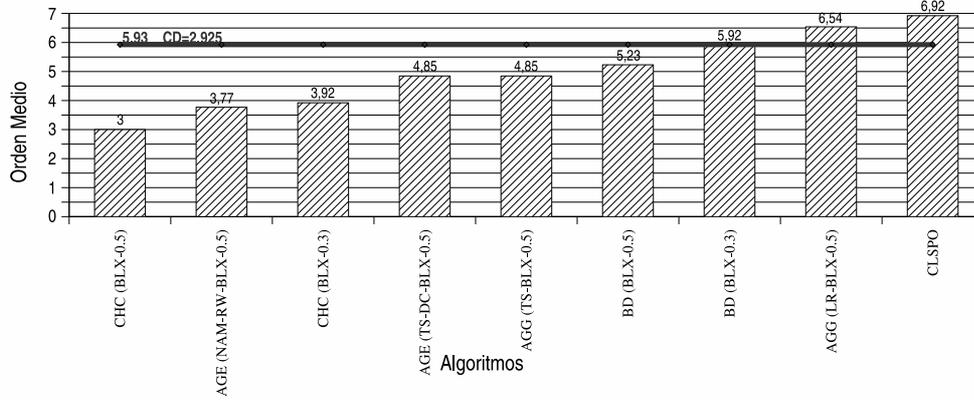
i	Algoritmo	z	p	$\alpha/i$	Aceptada o Rechazada
8	AGG (LR-BLX-0.5)	4.081	0.000044	0.0063	R
7	CLPSO	4.010	0.000060	0.0071	R
6	BD (BLX-0.3)	3.544	0.0003929	0.0083	R
5	BD (BLX-0.5)	2.219	0.0264217	0.0100	A
4	AGG (TS-BLX-0.5)	2.112	0.0346402	0.0125	A
3	CHC (BLX-0.3)	1.718	0.0856734	0.0167	A
2	AGE (TS-DC-BLX-0.5)	0.895	0.3707103	0.0250	A
1	CHC (BLX-0.5)	0.429	0.6674364	0.0500	A

Tabla 2.26: Test de Holm de los distintos AEs respecto al mejor: AGE (NAM-RW-BLX-0.5) (SW 50)

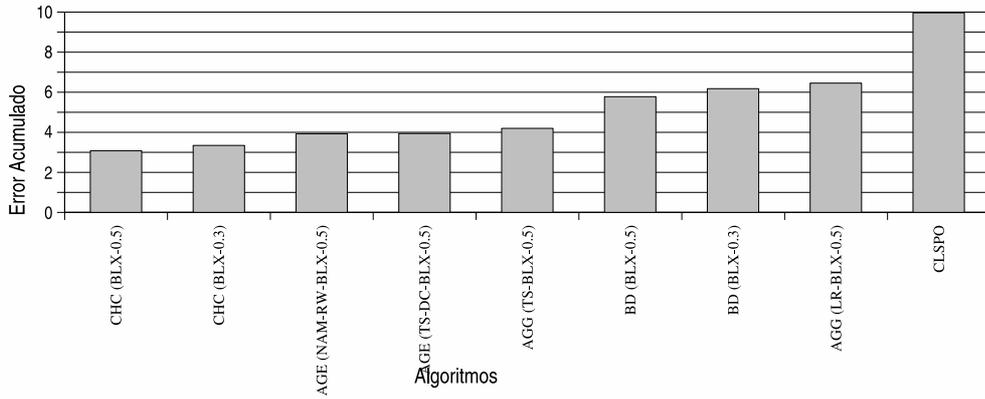
**AEs con SW 100** La Figura 2.36 muestra el resultado de comparar las dos mejores variantes de cada AE con SW 100. Se puede observar que:

- Las mejores propuestas son las que usan el enfoque de AGE y CHC. En particular, la mejor es *CHC (BLX-0.5)*. Dado que *CHC (BLX-0.5)* obtiene mejores resultados tanto en orden medio como en EAc, se puede afirmar que es el mejor modelo de AE para SW 100.
- Las tendencias comentadas en el punto anterior sobre CLPSO y BD se confirman.
- No se aprecia ninguna diferencia significativa entre los esquemas del AGE y CHC.

Hemos aplicado el operador de Holm. La Tabla 2.27 muestra los resultados, que confirma los resultados del test de Bonferroni-Dunn.



(a) Test de Bonferroni-Dunn



(b) Errores Acumulados

Figura 2.36: Comparación de las mejores variantes de los AEs, con Solis Wets y  $n_{int}=100$

i	Algoritmo	z	p	$\alpha/i$	Aceptada o Rechazada
8	CLPSO	3.652	0.000260	0.0063	R
7	AGG (LR-BLX-5)	3.294	0.000987	0.0071	R
6	BD (BLX-0.3)	2.721	0.006503	0.0083	R
5	BD (BLX-0.5)	2.076	0.037826	0.0100	A
4	AGG (TS-BLX-0.5)	1.718	0.085673	0.0125	A
3	AGE (TS-DC-BLX-0.5)	1.718	0.085673	0.0167	A
2	CHC (BLX-0.3)	0.859	0.390154	0.0250	A
1	AGE (NAM-RW-BLX-0.5)	0.716	0.473920	0.0500	A

Tabla 2.27: Test de Holm de los distintos AEs respecto al mejor: CHC (BLX-0.5) (SW 100)

### 2.3.4. Conclusiones de la Experimentación

De los análisis individuales de cada AE podemos resumir las siguientes conclusiones:

- El operador de cruce aritmético ofrece malos resultados. El operador de cruce  $BLX - \alpha$ , y en particular, BLX-0.5, ofrece mejores resultados.
- El operador de mutación BGA ha ofrecido mejores resultados que la mutación *No Uniforme*.
- El método de BL *Solis Wets* ha ofrecido mejores resultados que el método *Simplex*.
- Por su propio esquema de PSO, CLPSO no obtiene una mejora significativa al aplicar la BL aleatoriamente a individuos de su población. Aunque eso lo hace no recomendable para el modelo de hibridación más tradicional (aplicando una  $P_{LS}$ ), no tendría que ser así para otros esquemas de hibridación.
- El AGG produce mucha diferencia de comportamiento en función del operador de selección. En concreto, debido a su esquema de reemplazar una nueva población por la siguiente, los operadores de selección TS y LR, que son los que facilitan más que un buen individuo influya en las generaciones sucesivas, son las que ofrecen los mejores resultados.

- El AGE ofrece una gran variedad de comportamiento en función de la combinación de los operadores de selección y reemplazo empleados. Las combinaciones que mejores resultados ofrecen son aquellas que combinan en uno de los operadores diversidad y en el otro mayor presión selectiva, ya sea presión en la selección y diversidad en el reemplazo (*TS-DC*) o mayor diversidad en la selección y mayor presión selectiva en el reemplazo (*NAM-RW*).

Profundidad	Mejor Variante	Segunda Mejor
Sin BL	AGE (TS-DC-BLX-0.5)	AGE (NAM-RW-BLX-0.5)
SW 10	AGE (NAM-RW-BLX-0.5)	AGE (TS-DC-BLX-0.5)
SW 50	AGE (NAM-RW-BLX-0.5)	CHC (BLX-0.5)
SW 100	CHC (BLX-0.5)	AGE (NAM-RW-BLX-0.5)

Tabla 2.28: Resumen de las Mejores Variantes para cada valor de profundidad

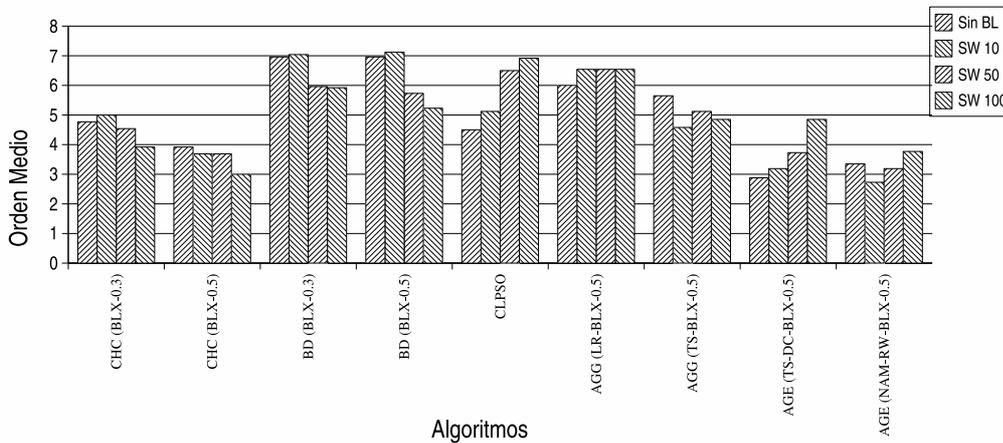


Figura 2.37: Orden Medio de los mejores AGEs, calculados por intensidad

La Tabla 2.28 muestra para cada intensidad los dos mejores algoritmos, y la Tabla 2.37 muestra para intensidad. De los resultados finales se pueden obtener las siguientes conclusiones:

- Se confirma que AGG, BD y CLPSO no son recomendables (son estadísticamente peores para ciertas intensidades).

- BD y CLPSO poseen tendencias contrarias conforme se incrementa la intensidad de la BL. La CLPSO presenta peores resultados conforme la intensidad de la BL aumenta, mientras que BD los resultados mejoran conforme aumenta ésta.
- Los algoritmos que presentan los mejores resultados son distintos para cada valor de intensidad. Sin embargo, destacan como algoritmos robustos el *AGE (NAM-RW-BLX-0.5)* y el *CHC (BLX-0.5)*
- No existe una diferencia significativa entre AGE y CHC. Sin embargo, es el modelo AGE es que presenta mejores resultados, y mayor robustez (en particular, el algoritmo *AGE (NAM-RW-BLX-0.5)*). Además, Whitley y Garret [WG03] afirman que el modelo CHC presenta dificultades para mejorar sus resultados mediante la incorporación de una BL. Por tanto, recomendamos el uso del modelo *AGE (NAM-RW-BLX-0.5)*, por lo que será el esquema empleado en los capítulos posteriores.
- Se ha confirmado que a la hora de estudiar la conveniencia de un AE u otro, es muy determinante la correcta combinación de los distintos operadores. Un AE al que se aplicase una inadecuada combinación de operadores (como el AGE) podría catalogarse muy rápidamente como inadecuado. Esto vuelve a confirmar que para elegir un AE es determinante una sabia elección de cada uno de sus componentes.

Este AE será el empleado en todos los modelos meméticos que propondremos en capítulos posteriores, ya que ha mostrado un buen comportamiento tanto sin aplicarle BL como aplicándosela.

## 2.4. Conclusiones

En este capítulo hemos abordado el problema de elegir un AE adecuado para su uso dentro de un AM. Dentro de esta problemática hemos planteado la necesidad de un estudio que comparase distintos AEs para distintos valores de intensidad de BL, y conocer así su comportamiento al trabajar conjuntamente con éstos. En este capítulo hemos procedido a un análisis con esas características.

El análisis se ha realizado con variantes meméticas que hacen uso de un conjunto pequeño pero representativo de AEs clásicos (CHC, BD, CLPSO, AGG y

AGE), y distintos operadores que definen su comportamiento. Dado que la posibilidad de elegir para cada tipo de operador (cruce, mutación, selección, estrategia de reemplazo, . . .) de entre un conjunto puede afectar fuertemente al modelo AE sobre el que se aplique, hemos definido sobre cada operador un cierto conjunto de posibilidades, y a la hora de valorar un modelo de AE hemos considerado el comportamiento de dicho AE sobre cada combinación posible de éstas. De esta forma, para cada AE hemos obtenido un cierto número de variantes, en algunos casos bastante alto.

Así pues, a la hora de comparar los distintos modelos hemos tenido que comparar, previamente, para cada AE, sus variantes respectivas, y hemos analizado principalmente: La influencia de los operadores considerados; cómo afecta la incorporación de la BL a los resultados; y cuáles eran las dos mejores variantes. Para garantizar la fiabilidad de las conclusiones obtenidas se han aplicado distintos tests estadísticos.

De los análisis individuales de cada AE podemos resumir las siguientes conclusiones:

- Los métodos de cruce y mutación que han obtenido mejores resultados han sido aquellos que producían mayor diversidad en la búsqueda, confirmando así la idea de que un AE debe de fomentar suficiente diversidad para su mejor uso dentro de un AM.
- La mejoría de incorporar la BL varía enormemente en función del AE considerado. Alguno (como CLPSO) no es mejorado por la incorporación de la BL. Otro, BD, sí obtiene una mejora muy significativa. Pero en general, la incorporación de la BL ofrece una mejora muy importante.
- La influencia de los operadores de selección y reemplazo es muy determinante. En el AGG, se obtienen mejores resultados con operadores de selección con alta presión selectiva (como TS), y en el AGE las combinaciones de selección y reemplazo que mejores resultados ofrecen son aquellas que combinan en uno de los operadores diversidad y en el otro mayor presión selectiva (como el uso combinado de NAM y RW).

Tras este primer estudio, hemos procedido a comparar las dos mejores variantes de cada modelo de AE entre sí, y hemos reflejado las conclusiones obtenidas. Lo más destacable es la gran diferencia entre usar un AE u otro. En particular, se observa que los modelos de AEs más recomendados son CHC y AGE (aplicando

los mejores operadores considerados en cada caso). Sin embargo, por su mayor robustez, elegimos como más recomendable el AGE con la siguiente configuración:

- Población de 60 individuos.
- Método de selección de padres NAM, con  $N_{NAM} = 3$ .
- Operador de cruce BLX-0.5.
- Operador de mutación BGA, al 12.5 % de los cromosomas.
- Estrategia de reemplazo RW.

Por tanto, en los capítulos posteriores procederemos a utilizar este AGE como base para nuestros propios modelos meméticos.

## Capítulo 3

# Algoritmo Memético con Búsqueda Local Basada en el Operador de Cruce ( $AMCR-BLC^2P$ )

Un tipo de AMCRs que actualmente está recibiendo especial atención es el formado por aquellos que utilizan procedimientos de BL basados en el operador de cruce (BLC) [DJ02],[DNVHW03],[SYK96],[NI05]. Dado que este operador produce hijos alrededor de los padres, puede considerarse como mecanismo para la generación de nuevas soluciones dentro de un método de BL. El diseño de estrategias de BLC es particularmente atractivo para el caso de los AMCRs, ya que existen operadores de cruce para codificación real que muestran un comportamiento de *autoadaptación* (generan los hijos en función de la distribución de los padres, sin necesidad de parámetros adicionales) [BD01]. Conforme avanzan las generaciones, la población de un AMCR pierde diversidad, favoreciendo la proximidad entre los padres. Bajo esta circunstancia, estos operadores de cruce generan hijos muy próximos a los padres, produciendo un adecuado ajuste local (propiedad pretendida precisamente para una buena técnica de BL). Así pues, este tipo de operadores representa un camino prometedor hacia el diseño de procedimientos de BLC efectivos.

En este capítulo presentamos un operador de cruce centrado en un padre (llamado  $PBC - \alpha$ ), especialmente diseñado para la construcción de un método de BL basado en el cruce.

También presentamos una BLC denominada  $BLC^2P$  que emplea el operador de cruce  $PBC - \alpha$ .  $BLC^2P$  es un tipo particular de BLC diseñado para aprovechar la capacidad de autoadaptación de los operadores de cruce para codificación real, constituyendo así un método de BLC *autoadaptable*. El objetivo de  $BLC^2P$  es obtener los mejores niveles de *precisión* posible para conducir la población hacia las áreas del espacio más prometedoras, produciendo un refinamiento eficaz sobre ellas.

Por otro lado, presentamos un AMCR diseñado para promover altos niveles de diversidad en la población que hace uso de  $BLC^2P$  para mejorar las soluciones encontradas. Intenta fomentar la *fiabilidad* del proceso de búsqueda asegurando que la BL se centrará sobre diferentes zonas prometedoras a lo largo de la ejecución. Además, el AMCR emplea un mecanismo adaptable que determina la probabilidad con la cual cada solución creada por el proceso genético será punto de partida de dicha BL. De esta forma, este algoritmo ajusta la relación entre búsqueda global y búsqueda local (es decir, la balanza entre exploración y explotación) a las características particulares del problema a resolver.

El capítulo queda estructurado de la siguiente forma. En la Sección 3.1 revisamos algunos aspectos importantes sobre los operadores de cruce para codificación real y presentamos el utilizado en este capítulo. En la Sección 3.2, describimos nuestra propuesta de BL. En la Sección 3.3 presentamos la propuesta de AMCR que hace uso de dicha BL, poniendo especial énfasis en el mecanismo de hibridación empleado. En la Sección 3.4, se detalla el marco experimental de los experimentos y las funciones de comparación empleadas. En la Sección 3.5, se realiza un estudio experimental de los distintos componentes del AMCR propuesto: la influencia de la intensidad de la BL, la sinergia existente entre el  $BLC^2P$  y el  $NAM$ , y el mecanismo autoadaptable de la BL y su conveniencia. En la Sección 3.6, se hace un estudio comparativo de la propuesta, frente a otros modelos de AEs. Finalmente, en el apartado 3.7 se presentan las conclusiones finales del capítulo.

### 3.1. Operador de Cruce BLX – $\alpha$ Centrado en los Padres: PBC – $\alpha$

El operador de cruce es un operador fundamental para los AGs [JS92],[Kit01], ya que explota la información sobre el espacio de búsqueda disponible en la población. El desarrollo de operadores de cruce sofisticados para AGCRs ha supuesto un gran esfuerzo dentro de la investigación de estos algoritmos, y como resultado, han surgido distintas propuestas [HLV98],[Deb01],[HLS03]. La codificación real de las soluciones ofrece la posibilidad de definir una amplia gama de operadores de cruce para AGCRs que pueden tomar ventaja de su naturaliza numérica. Muchos de estos operadores se definen a partir de una distribución de probabilidad para los hijos basándose en alguna medida de distancia entre las padres. Si los padres están muy próximos, los hijos generados en el operador de cruce van a estar densamente distribuidos alrededor de los padres, produciéndose un ajuste efectivo o local. Por el contrario, si los padres están distantes entre sí, entonces los hijos estarán repartidos en un área amplia del espacio de búsqueda. De esta forma, estos operadores permiten que se produzca la convergencia, divergencia y adaptación a funciones objetivo cambiantes, sin la necesidad de un parámetro extra o de un mecanismo especial para alcanzar este fin.

De hecho, recientemente, se ha demostrado que los AGCRs con alguno de estos operadores exhiben un comportamiento autoadaptable similar al observado en las estrategias de evolución y programación evolutiva [BD01],[Kit01]. Es más, Beyer y Deb [BD01] mantienen que la utilización de la distancia entre padres para la definición de un operador de variación es una característica imprescindible para que un algoritmo evolutivo que use dicho operador manifieste un comportamiento autoadaptable.

En este capítulo se propone un nuevo operador de cruce específicamente diseñado para construir una BLC, extendiendo el operador BLX- $\alpha$  presentado por Eshelman y Schaffer [ES93]. Lo llamamos operador BLX- $\alpha$  *centrado en los padres* (PBC- $\alpha$ ) y se describe a continuación.

Consideremos que  $X = (x_1 \dots x_n)$  y  $Y = (y_1 \dots y_n)$  ( $x_i, y_i \in [a_i, b_i] \subset \mathbb{R}$ ,  $i = 1 \dots n$ ) son dos cromosomas con codificación real seleccionados para aplicarles el operador de cruce.

PBC- $\alpha$  genera uno de estos dos posibles hijos aleatoriamente,  $Z_1 = (z_1^1 \dots z_n^1)$  ó  $Z_2 = (z_1^2 \dots z_n^2)$ , donde  $z_i^1$  es un número aleatorio elegido en el intervalo  $[x_i - I \cdot \alpha, x_i + I \cdot \alpha]$  y  $z_i^2$  en el intervalo  $[y_i - I \cdot \alpha, y_i + I \cdot \alpha]$ , en donde  $I = |x_i - y_i|$ .

La Figura 3.1 muestra un ejemplo de aplicación del  $PBC - \alpha$ .

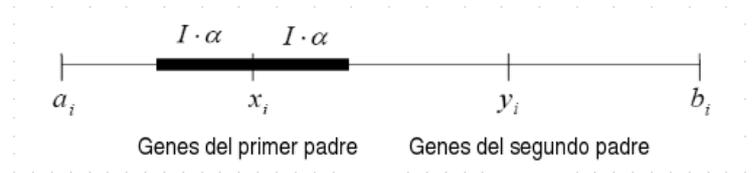


Figura 3.1: Operador de cruce  $PBC - \alpha$

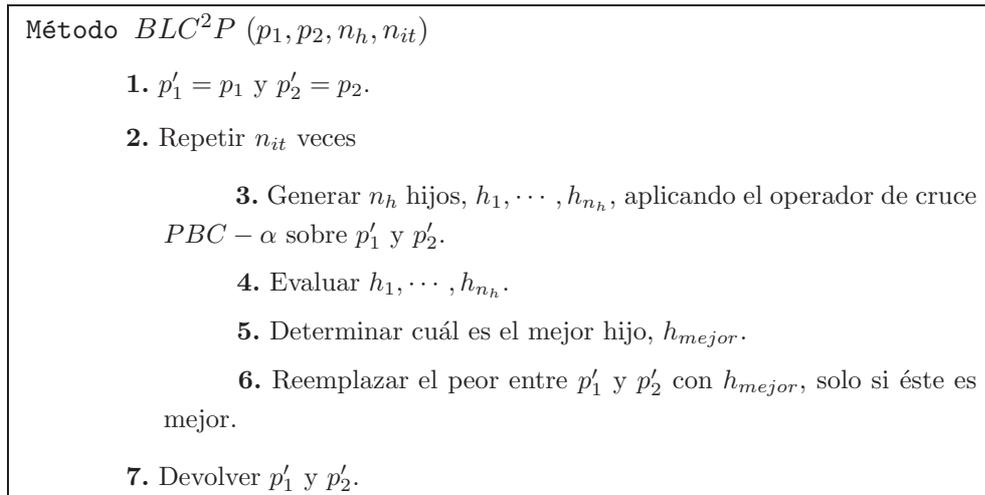
Este operador muestra las siguientes características:

- Da alta prioridad a la creación de hijos próximos a sus padres. Estudios realizados por Deb, Anand y Joshi [DAJ02] han mostrado que los operadores centrados en los padres resultan ser un mecanismo significativamente efectivo para resolver problemas de optimización continua.
- Promueve niveles de diversidad en la población que pueden ajustarse fácilmente variando su parámetro asociado  $\alpha$ . Cuanto más alto sea  $\alpha$ , más se favorecerá la diversidad en la población.
- Crea hijos atendiendo a la distancia entre los padres. Por ello, un AGCR que lo use potencialmente podrá exhibir autoadaptación.

### 3.2. Método de Búsqueda Local Continuo Basado en el operador $PBC - \alpha$ : $BLC^2P$

En esta sección describimos el algoritmo de BLC propuesto, denominado *Algoritmo de Búsqueda Local Basado en un Cruce Centrado en el Padre,  $BLC^2P$* .

$BLC^2P$  sigue el mismo esquema que un algoritmo de ascensión de colinas. En un algoritmo de ascensión de colinas se parte de una única solución y, en cada paso, se genera una nueva solución candidata utilizando un operador de movimiento. El algoritmo repite en cada iteración el proceso de búsqueda partiendo de la solución candidata en vez de la actual si la candidata posee un valor mejor o igual para la función objetivo. Jones [Jon95] y O'Reilly y Oppacher [OO95]

Figura 3.2: Esquema de  $BLC^2P$ 

fueron los primeros en describir técnicas de búsqueda local basadas en el operador de cruce. La idea básica consiste en utilizar un algoritmo de ascensión de colinas como mecanismo de búsqueda y el operador de cruce como operador de movimiento.

El operador  $BLC^2P$  mantiene un par de padres y repetidamente le aplica el cruce hasta generar un número determinado de hijos,  $n_h$ . Después, el mejor hijo reemplaza al peor de los padres, siempre y cuando tenga un valor de función objetivo mejor. Este proceso se repite durante  $n_{it}$  iteraciones y devuelve los dos padres finales. Para usar  $BLC^2P$ , necesitamos un valor para los parámetros  $n_h$  y  $n_{it}$  y un par de padres inicial,  $(p_1, p_2)$ . La Figura 3.2 muestra en pseudocódigo el algoritmo para el propuesto  $BLC^2P$ . Para llevar a cabo el paso 3, proponemos usar el propuesto  $PBC - \alpha$  como operador de cruce.

Una idea básica al definir este modelo es explotar el comportamiento auto-adaptativo del cruce dentro del propio procedimiento, produciéndose así el diseño de una BL *auto-adaptativa*. El proceso competitivo (producido por el paso 6) modifica, durante la ejecución de la BL el par de padres actuales, cambiando así la distancia entre ellos, diferencia fundamental que lo distingue de una simple aplicación de cruce con múltiples descendientes. Dado que los operadores de cruce para codificación real generan hijos en función de la distribución existente entre

los padres, la convergencia o divergencia de la BL se lleva a cabo sin requerir un parámetro adaptable adicional.

La mayoría de las técnicas de BLs continuas, tales como el algoritmo Solis Wets [SW81] y la estrategia de evolución (1+1) hacen uso de un parámetro de control externo (*tamaño de paso*) para guiar la búsqueda. Además, adaptan este parámetro para realizar movimientos de diferente amplitud en función del éxito de los pasos anteriores. Este tipo de técnicas consideran una adaptación *explícita* de parámetros, frente a la autoadaptación de BLC<sup>2</sup>P.

La idea de emplear modelos AGs para funcionar como BL no es nuevo, en [KPTP01] se propone el uso de un algoritmo microgenético, que es un AG con una pequeña población que evoluciona durante un periodo pequeño de generaciones, como un operador de BL genérico. BLC<sup>2</sup>P puede considerarse un algoritmo microgenético que utiliza la menor población posible para aplicar el proceso de BL, de únicamente dos cromosomas.

### 3.3. Propuesta: AMCR-BLC<sup>2</sup>P

En esta sección, presentamos nuestra propuesta de AMCR basado en el uso de BLC<sup>2</sup>P presentada en la Sección 3.2, al que llamaremos AMCR-BLC<sup>2</sup>P:

- La misión del BLC<sup>2</sup>P es obtener los mejores niveles de *precisión* posibles para conducir a la población hacia las zonas de búsqueda más prometedoras, produciendo un refinamiento efectivo sobre ellas.
- El AMCR está diseñado para promover niveles de diversidad altos en la población. Con ello, intenta fomentar la *fiabilidad* del proceso de búsqueda, asegurando que BLC<sup>2</sup>P se centrará sobre diferentes zonas prometedoras a lo largo de la ejecución.

Esta sección se estructura de la siguiente forma:

- En la Sección 3.3.1, explicamos los pasos que constituyen el AMCR (el esquema general).
- En la Sección 3.3.2, presentamos el mecanismo de hibridación presentado, un mecanismo adaptable que asigna a cada cromosoma, en función de sus características, una probabilidad de ser refinado mediante la BL propuesta.

### 3.3.1. Esquema AMCR general

En esta Sección, presentamos nuestra propuesta de AMCR basado en el uso de  $BLC^2P$ . Las principales características del AMCR propuesto son (ver Figura 3.3):

1. Inicializar población.
2. Mientras (*no condición-terminación*) hacer
  3. Usar Estrategia-emparejamiento-variada para seleccionar dos padres.
  4. Aplicar  $PBC - \alpha$  y mutación-BGA para crear un hijo,  $h_{nuevo}$ .
  5. Evaluar  $h_{nuevo}$ .
  6. Aplicar Mecanismo-adaptable- $p_{BL}$  para obtener la  $p_{BL}$  para  $h_{nuevo}$ .
  7. Si  $u(0, 1) < p_{BL}$  entonces
    8. Encontrar el mejor cromosoma en la población,  $c_{mejor}$ .
    9. Aplicar  $BLC^2P(h_{nuevo}, c_{mejor}, n_h, n_{it})$ .  
Se devuelven  $c_{acc}^1$  y  $c_{acc}^2$  ( $c_{acc}^1$  es el mejor).
    10. Reemplazar  $c_{mejor}$  con  $c_{acc}^1$ , sólo si lo mejora.
    11. Utilizar Estrategia-reemplazo-estándar para insertar  $c_{acc}^2$  en la población.
12. Sino
  13. Emplear Estrategia-reemplazo-estándar para insertar  $h_{nuevo}$  en la población.

Figura 3.3: AMCR estacionario propuesto

- Explota la información que pueden aportar los padres a partir del concepto de herencia asociado al operador de cruce.
- El modelo propuesto utiliza el modelo evolutivo resultante del Capítulo 2, el AGE con la estrategia de reemplazo RW y el método de selección

*NAM*. Además, para introducir mayor diversidad se aplicará el operador de mutación BGA [MSV93]. Se puede consultar el Capítulo 2 para obtener mayor información.

- Como operador de cruce, para fomentar un alto nivel de diversidad en la población, se utiliza el operador de cruce  $PBC - \alpha$  propuesto (ver Sección 3.1), con un alto valor para su parámetro asociado ( $\alpha = 1$ ).
- Emplea el método  $BLC^2P$  presentado en la Sección 3.2 para refinar los nuevos cromosomas creados mediante el operador de cruce y el método de mutación.  $BLC^2P$  requiere dos padres iniciales para comenzar su operación: el primero será el nuevo cromosoma generado, y el segundo será el mejor elemento presente actualmente en la población. Tras su aplicación, devuelve dos soluciones. La mejor puede que mejore al mejor elemento actual en la población. Cuando esto ocurra, se incluirá en la población y el antiguo mejor se borra. La otra solución se introducirá en la población siguiendo la *estrategia de reemplazo estándar*. Esta estrategia consiste en sustituir el peor individuo de la población, siempre y cuando el cromosoma a insertar sea mejor. Es destacable la alta capacidad de explotación, que conduce a una convergencia muy rápida. Esta tendencia se intenta contrarrestar mediante el conjunto de mecanismos aplicados para fomentar diversidad (cruce empleado, selección *NAM*, ...).
- Utiliza el mecanismo adaptable explicado en la sección siguiente, que asigna a cada cromosoma una probabilidad de ser refinado mediante la BL. A este parámetro lo notaremos como  $p_{BL}$ . Los cromosomas que mejoren al peor individuo de la población se les aplicará la máxima  $p_{BL}$  ( $p_{BL}=1$ ), por lo tanto serán siempre mejorados siguiendo el proceso  $BLC^2P$ . A los cromosomas que no cumplan la condición anterior se les aplicará el operador de cruce con una pequeña valor de  $p_{BL}$ ,  $p_{BL} = 0.0625$ , que es el valor recomendado por varios autores [Har94],[HRBM00] y que se empleó en el capítulo anterior.
- La intensidad de la BL está determinado por los parámetros  $n_{it}$  y  $n_h$  ( $n_{int} = n_h \cdot n_{it}$ ).
- La razón entre búsqueda local y global ( $\frac{L}{G}$ ) está determinada por los parámetros  $n_{it}$ ,  $n_h$  y  $p_{BL}$ , ya que cuanto mayor sean estos parámetros mayor será la relación  $\frac{L}{G}$ . La ratio  $\frac{L}{G}$  determina el equilibrio entre las capacidades de exploración del AMCR, y las habilidades explotativas de  $BLC^2P$ , por lo

que influye de forma importante en el rendimiento final del algoritmo para un problema concreto. Cuanto mayor sean los valores de dichos parámetros, más cercano al 100 % estará la ratio  $\frac{L}{G}$ . Para problemas complejos, una baja ratio puede ser más efectiva, al favorecer la exploración mediante mayor búsqueda global; mientras que para problemas menos complejos es más adecuado un mayor valor  $\frac{L}{G}$ , para permitir a la BL explotar el espacio de búsqueda durante un mayor tiempo, aprovechando más su capacidad de refinar soluciones. Con el empleo del mecanismo  $p_{BL}$  adaptativo, deseamos que se ajuste (en la medida de lo posible) dicho valor  $\frac{L}{G}$  al valor más adecuado en cada caso al problema considerado.

### 3.3.2. Mecanismo Adaptable para Determinar $p_{BL}$

Normalmente, la BL se aplica sobre una pequeña porción del total de soluciones visitadas, ya que el número de evaluaciones de la función objetivo requeridas por la BL puede llegar a ser excesivo. Ante este hecho surge una cuestión importante: cómo seleccionar las soluciones sobre las cuales se aplicará la BL. La propuesta clásica, ya comentada, es la de aplicar una probabilidad fija  $p_{BL}$  [Har94]. Sin embargo, el uso de una probabilidad independiente del individuo a considerar presenta una serie de desventajas:

- Las soluciones con mejor valor para la función objetivo presentan la misma probabilidad de ser mejoradas por la BL que el resto, a pesar de que, por su buen comportamiento, serían soluciones que merecerían ser explotadas en su entorno cercano.
- El esfuerzo invertido en aplica la BL está determinado por la intensidad de aplicación de la BL, y con el número de individuos sobre los que aplicar. Dado que, para ser efectiva, cada BL puede requerir un cierto nivel de intensidad diferente (como observamos en el Capítulo 2), el margen de maniobra sobre dicho parámetro es escaso, con lo que el número de soluciones que puede ser mejorados por la BL suele ser muy reducido. Por tanto, una elección aleatoria no garantiza que sea aplicada a buenas soluciones.

Al estudiar estos inconvenientes se observó que la BL empleada,  $BLC^2P$ , conducía a una rápida convergencia, por lo que sería inadecuado aplicarlo de forma totalmente aleatoria, ya que la mejora obtenida al aplicarse sobre soluciones

no prometedoras podría no compensar la pérdida de diversidad que produciría. La idea de una aplicación adaptable de BL no es nueva, otros autores han propuesto, como solución, propuestas de AM adaptables [OLZW06]. En nuestro caso, hemos considerando el esquema estacionario de nuestro AM, para poder proponer un esquema adaptable bastante simple para asignar una probabilidad de BL a cada cromosoma generado mediante cruce y mutación,  $c_{nuevo}$ :

$$p_{BL} = \begin{cases} 1 & \text{Si } f(h_{nuevo}) \text{ es mejor que } f(c_{peor}) \\ 0,0625 & \text{Otro caso} \end{cases}$$

donde  $f$  es la función objetivo y  $c_{peor}$  es el peor elemento que está presente actualmente en la población.

Los cromosomas que mejoren al peor individuo de la población se les aplicará la máxima  $p_{BL}$  ( $p_{BL}=1$ ), por lo tanto serán siempre mejorados por la BL. A los cromosomas que no cumplan la condición anterior se les aplicará el operador de cruce con una pequeña valor de  $p_{BL}$ ,  $p_{BL} = 0.0625$ , que es el valor recomendado por varios autores [Har94, HRBM00] y que se empleó en el capítulo anterior.

Para el diseño de este método hemos considerado que un nuevo individuo,  $c_{nuevo}$ , que mejora al peor elemento actual es un elemento prometedor, y merece recibir un ajuste local. Por esta razón, nuestra aproximación adaptable asegura que el método  $BLC^2P$  se aplica sobre él. Además, el cromosoma resultante formará, con seguridad, parte de la población (ya que se usa la estrategia de reemplazo estándar). De esta forma, el AMCR estacionario conserva aquellos cromosomas que proporcionan una información concisa sobre la calidad de las mejores zonas del espacio de búsqueda. Por otro lado, cuando el nuevo cromosoma no cumple este requisito, se le asigna un valor bajo para la probabilidad de aplicación de la BL.

La ratio  $\frac{L}{G}$  determina el equilibrio entre la habilidad para explorar del AMCR y la capacidad de explotación de  $BLC^2P$ , y por tanto, tiene una influencia determinante sobre la efectividad el algoritmo para un problema particular. Para problemas complicados, lo más efectivo es mantener un valor bajo, ya que así se favorece la exploración (búsqueda global). Para problemas fáciles, lo que interesa es tener altos valores para la ratio, haciendo que  $BLC^2P$  explote durante mucho tiempo el espacio de búsqueda, y tome ventaja de su habilidad para refinar soluciones. Al emplear un mecanismo adaptable para obtener el valor de  $p_{BL}$ , intentamos ajustar (lo mejor posible) la ratio  $\frac{L}{G}$  a los valores que permitan alcanzar

una operación adecuada para el problema particular a resolver.

### 3.4. Marco Experimental

En esta sección se detalla el marco experimental llevado a cabo con el ánimo de determinar la efectividad del AMCR presentado en la Sección 3.3 y estudiar sus principales características.

Los problemas de prueba usados para los experimentos consisten en la minimización de seis funciones y tres problemas reales. Las funciones, ya empleadas en el capítulo anterior, son las siguientes:

- Modelo *esférico* ( $f_{Esf}$ ) [dJ75, Sch81].
- Función de *Rosenbrock* generalizada ( $f_{Ros}$ ) [dJ75].
- Problema de *Schwefel 1.2* ( $f_{Sch}$ ) [Sch81].
- Función de *Rastrigin* generalizada ( $f_{Ras}$ ) [B92, TA89].
- Función de *Griewangk* ( $f_{Gri}$ ) [Gri81].

La dimensión del espacio de búsqueda es 25 para todas ellas.  $f_{Esf}$ ,  $f_{Ros}$  y  $f_{Sch}$  son funciones continuas y unimodales, mientras que  $f_{Ras}$  y  $f_{Gri}$  son continuas y multimodales.

Los problemas reales son:

- *Sistema de ecuaciones lineales* ( $P_{sle}$ ) [EMS97].
- *Identificación de parámetros de la modulación de la frecuencia de sonido* ( $P_{fms}$ ) [TF93].
- *Aproximación polinomial* ( $P_{Cheb}$ ) [SP95].

El valor de la función objetivo del óptimo de todos los problemas de prueba es cero.

Comparando los distintos algoritmos o variantes entre sí se ha aplicado el test de Student, también conocido como *t-test*[Gos08]. Este test es uno de los test estadísticos más empleados en la práctica para comparar dos grupos independientes de observaciones respecto a una variable numérica.

Sean  $\bar{X} = X_1, X_2, \dots, X_n$  y  $\bar{Y} = Y_1, Y_2, \dots, Y_n$  el conjunto de resultado de dos variables (o algoritmos) de tamaño  $n$ , se define la variable *t-test* según la ecuación 3.1.

$$T_{test} = \frac{\text{media}(\bar{X}) - \text{media}(\bar{Y})}{\sqrt{\sigma_X^2 + \sigma_Y^2}} \quad (3.1)$$

En donde  $\text{media}(\bar{X}), \text{media}(\bar{Y})$  representa la media de los resultados de las variables  $\bar{X}$  e  $\bar{Y}$ , respectivamente.  $\sigma_X$  y  $\sigma_Y$  representan la desviación típica de los resultados de  $X$  e  $Y$ , respectivamente.

### 3.5. Análisis de los Componentes

En este apartado analizamos nuestra propuesta para estudiar cómo influye cada parámetro e identificar sus valores más adecuados. Este apartado posee la siguiente estructura:

- En la Sección 3.5.1, se realizan varios experimentos para determinar un valor adecuado de intensidad.
- En la Sección 3.5.2, se estudia la conveniencia de emplear tanto la BL propuesta, *BLC<sup>2</sup>P*, como el método de selección *NAM*.
- En la Sección 3.5.3, se estudia la conveniencia del criterio autoadaptable de la BL (ver Sección 3.2).
- En la Sección 3.5.4, se estudia si efectivamente se adapta la ratio  $\frac{L}{G}$  a cada problema, y la conveniencia del criterio adaptable de *pBL*.

#### 3.5.1. Influencia de la Intensidad

En este primer estudio investigamos la influencia de la intensidad sobre la efectividad del AMCR propuesto, por tener un efecto significativo sobre la ratio

$\frac{L}{G}$ . En particular, analizamos el comportamiento del algoritmo usando diferentes valores para  $n_{it}$ , manteniendo un valor fijo para  $n_h$  (el otro parámetro asociado con  $BLC^2P$ , el cual especifica el número de hijos generados a partir del par de padres actual) ( $n_h = 3$ ).

Hemos implementado un AMCR estacionario que aplica  $BLC^2P$  y utiliza el operador  $PBC - \alpha$  ( $\alpha = 1$ ) (Sección 3.1). También emplea el operador de mutación BGA [MSV93] con una probabilidad de  $\frac{1}{n}$ . El tamaño de la población es de 60 cromosomas y hemos considerado un valor bastante alto para el parámetro asociado al emparejamiento variado,  $N_{NAM} = 25$ . El algoritmo se ha ejecutado 50 veces con 100.000 evaluaciones cada una. Las medidas de eficacia utilizadas son las que se detallan a continuación.

- Medida **A** : media del mejor valor de la función objetivo encontrada al final de cada ejecución.
- Medida **B**: mejor valor de la función objetivo obtenido en las 50 ejecuciones. En el caso de que el valor óptimo (un valor de cero para la función objetivo) se localice durante alguna ejecución, esta medida reflejará el porcentaje de ejecuciones en las que esto ocurra.

La Tabla 3.1 muestra los resultados obtenidos para distintos valores de  $n_{it}$  (resaltando en negrita los mejores valores). Podemos observar que los mejores valores para la medida *A*, para cada problema, se alcanza con distintos valores para  $n_{it}$ , como era de esperar:

- Para las funciones de prueba multimodales,  $f_{Ras}$  y  $f_{Gri}$ , y para todos los problemas reales (los cuales son bastante complejos), los mejores resultados se alcanzan usando valores bajos para  $n_{it}$ . Estos valores bajos favorecen la búsqueda global a través del AMCR, propiciando niveles altos de diversidad. Esta conducta es esencial para manejar este tipo de problemas.
- Los valores altos para  $n_{it}$  proporcionan una operación prolongada para la BL, favoreciendo un refinamiento local efectivo sobre las funciones de prueba unimodales,  $f_{Esf}$ ,  $f_{Ros}$  y  $f_{Sch}$ .

Hemos escogido un valor particular para  $n_{it}$ , para poder proseguir con el estudio de nuestra propuesta y poder compararla con otros modelos de AMs. Consideramos que con  $n_{it} = 3$ , se alcanza un nivel de robustez aceptable con

$n_{it}$	$f_{Esf}$		$f_{Ros}$		$f_{Sch}$		$f_{Ras}$	
	A	B	A	B	A	B	A	B
1	1.5e-045	2.2e-049	1.1e+001	9.3e-002	8.5e+001	6.4e+000	<b>8.9e-001</b>	40.0 %
2	1.4e-080	2.0e-085	3.8e+000	5.6e-005	6.5e-004	2.2e-005	9.2e-001	<b>46.0 %</b>
3	6.5e-101	1.1e-105	2.2e+000	6.0e-004	3.8e-007	4.5e-009	1.4e+000	32.0 %
4	2.6e-110	1.1e-116	<b>1.4e+000</b>	<b>8.6e-007</b>	1.7e-008	1.1e-010	1.8e+000	22.0 %
5	4.5e-117	3.1e-123	2.8e+000	1.4e-005	2.8e-007	2.4e-012	1.1e+000	26.0 %
6	1.1e-118	<b>3.2e-125</b>	1.8e+000	2.2e-005	1.3e-009	2.9e-012	1.4e+000	20.0 %
7	<b>5.2e-119</b>	1.2e-124	1.7e+000	1.3e-004	1.1e-009	1.5e-012	1.9e+000	20.0 %
8	1.1e-115	4.1e-123	3.6e+000	6.3e-005	2.7e-004	1.0e-012	1.7e+000	18.0 %
9	3.6e-113	2.7e-119	2.5e+000	5.1e-005	1.2e-008	3.1e-013	2.7e+000	8.0 %
10	1.0e-106	7.2e-116	5.1e+000	2.4e-004	<b>1.2e-010</b>	<b>1.2e-013</b>	2.4e+000	12.0 %
15	2.8e-081	8.2e-091	6.2e+000	8.9e-003	6.0e-008	6.4e-012	3.0e+000	2.8e-014
20	3.2e-063	1.5e-070	1.4e+001	4.9e-002	2.0e-007	1.6e-009	6.9e+000	3.9e-011
50	4.8e-026	1.7e-030	2.0e+001	2.8e+000	6.9e-002	5.6e-004	2.2e+001	7.0e+000

$n_{it}$	$f_{Gri}$		$P_{ste}$		$P_{Cheb}$		$P_{fms}$	
	A	B	A	B	A	B	A	B
1	<b>5.6e-003</b>	<b>58.0 %</b>	5.2e+001	1.8e+000	1.3e+003	6.4e+001	<b>4.1e+000</b>	<b>42.0 %</b>
2	1.3e-002	28.0 %	<b>2.7e+001</b>	2.4e+000	2.1e+002	5.3e+000	7.1e+000	36.0 %
3	1.3e-002	30.0 %	5.5e+001	7.9e-001	<b>1.4e+002</b>	9.2e+000	7.7e+000	40.0 %
4	2.2e-002	18.0 %	9.0e+001	3.5e+000	1.7e+002	1.7e+000	1.1e+001	18.0 %
5	1.8e-002	18.0 %	1.0e+002	2.6e+000	2.4e+002	1.2e+001	1.2e+001	24.0 %
6	1.7e-002	22.0 %	1.3e+002	7.7e+000	2.7e+002	1.2e+001	1.4e+001	6.0 %
7	2.2e-002	24.0 %	1.2e+002	4.7e+000	2.3e+002	2.5e+000	1.3e+001	10.0 %
8	2.4e-002	30.0 %	1.4e+002	1.5e+001	2.5e+002	9.3e+000	1.5e+001	4.0 %
9	2.8e-002	16.0 %	1.2e+002	7.3e+000	2.7e+002	1.2e+001	1.6e+001	6.0 %
10	2.6e-002	20.0 %	1.2e+002	1.6e+001	2.6e+002	3.3e+000	1.4e+001	16.0 %
15	3.0e-002	14.0 %	1.4e+002	1.0e+001	2.5e+002	2.3e+000	1.4e+001	18.0 %
20	2.9e-002	24.0 %	1.3e+002	<b>1.1e+000</b>	2.6e+002	1.2e+000	1.5e+001	10.0 %
50	3.9e-002	10.0 %	1.4e+002	2.1e+001	4.1e+002	<b>1.1e+000</b>	1.7e+001	4.0 %

Tabla 3.1: Resultados de AMCR-BLC<sup>2</sup>P con diferentes valores para  $n_{it}$

respecto a los otros valores para  $n_{it}$  analizados (véase la Tabla 3.1). En muchos casos, este valor ofrece unos resultados similares a los mejores obtenidos.

Es destacable que la mejor intensidad aplicada por aplicación de BL es relativamente baja, de sólo 9 individuos. Es un valor muy bajo si consideramos el requerido por otros métodos, como Simplex y Solis Wets. La capacidad de obtener resultados adecuados con una baja intensidad es una característica muy importante para su uso en AMCRs, ya que permite obtener buenos resultados permitiendo mantener, al mismo tiempo, un alto número de iteraciones en el proceso de búsqueda global.

### 3.5.2. Estudio de Sinergia entre BLC<sup>2</sup>P y NAM

Existen dos factores muy importantes en nuestro modelo, el método de búsqueda local empleado (BLC<sup>2</sup>P) que fomenta la explotación y el uso del empareja-

miento variado (*NAM*) en el cruce.

En esta sección intentamos determinar si ambos elementos se complementan de forma adecuada, es decir, si existe sinergia entre ambos. La sinergia ocurre si la combinación de los factores se comporta mejor que el uso aislado de cada uno de ellos [YM02].

Para estudiar si un factor es relevante, aplicamos el t-test, con un valor de 0.05 de significancia, para determinar si las diferencias encontradas son significativas respecto al resultado combinado.

Junto a los valores de bondad se añade un símbolo para indicar la dirección en función del resultado obtenido por el t-test:

- Un signo positivo (+) indica una mejora en el resultado de A (de la combinación de ambos factores respecto a la elección escogida).
  
- Un signo negativo (-) indica un empeoramiento.
  
- La no aparición de un signo indica que la diferencia encontrada no es significativa.

La Tabla 3.2 muestra el resultado de aplicar el t-test. Los resultados muestran que la elección propuesta (AMCR con *NAM* y *BLC<sup>2</sup>P*) mejoran de forma clara los resultados respecto a las otras alternativas consideradas. Se observa que el uso de *NAM* aparece tanto en el AE con los mejores resultados (con *BLC<sup>2</sup>P*), como en el AE con los peores (sin aplicarle la BL). Esto es debido a que en este último caso el nivel de diversidad ofrecido por *NAM* es excesivo, mientras que el uso de *NAM* cuando se aplica la BL es beneficiosa para contarrestar lo que podría ser, en otro caso, un exceso de explotación en la búsqueda. Por tanto, se confirma que el uso de la *BLC<sup>2</sup>P* permite mejorar los resultados, pero para ello se requiere mayor diversidad en la exploración.

Resumiendo, se puede observar que el uso de *NAM* es complementario al uso de *BLC<sup>2</sup>P* porque permite refinar los resultados al añadir mayor diversidad en la búsqueda. La combinación de ambos factores produce un equilibrio entre exploración y explotación que es decisivo para el éxito del algoritmo.

AMCR	$f_{Esf}$		$f_{Ros}$	
	A	B	A	B
sin NAM ni BLC <sup>2</sup> P	2.6e-023+	5.1e-025	2.2e+01+	1.9e+01
sin BLC <sup>2</sup> P	2.0e-016+	3.1e-017	2.0e+01+	1.8e+01
sin NAM	3.1e-114-	5.6e-120	1.9e+00	1.1e-04
con NAM y BLC <sup>2</sup> P	6.5e-101	1.1e-105	2.2e+00	6.0e-04
AMCR	$f_{Sch}$		$f_{Ras}$	
	A	B	A	B
sin NAM ni BLC <sup>2</sup> P	3.4e-01+	4.5e-02	1.0e+00	12.0%
sin BLC <sup>2</sup> P	5.7e+02+	2.3e+02	3.1e+00+	1.1e-7
sin NAM	3.3e-05	9.7e-14	2.1e+00	8.0%
con NAM y BLC <sup>2</sup> P	3.8e-07	4.5e-09	1.4e+00	32.0%
AMCR	$f_{Gri}$		$P_{sle}$	
	A	B	A	B
sin NAM ni BLC <sup>2</sup> P	4.0e-03-	74.0%	7.6e+01	3.4e+00
sin BLC <sup>2</sup> P	1.9e-02	3.1e-014	3.4e+02+	1.2e+02
sin NAM	2.6e-02+	18.0%	1.7e+02+	2.0e+00
con NAM y BLC <sup>2</sup> P	1.3e-02	30.0%	5.5e+01	7.9e-01
AMCR	$P_{Cheb}$		$P_{fms}$	
	A	B	A	B
sin NAM ni BLC <sup>2</sup> P	3.9e+02+	2.5e+01	9.0e+00	8.0%
sin BLC <sup>2</sup> P	2.6e+03+	4.6e+02	1.5e+01+	1.2e+01
sin NAM	5.0e+02+	4.3e+01	1.5e+01+	8.0%
con NAM y BLC <sup>2</sup> P	1.4e+02	9.2e+00	7.7e+00	40.0%

Tabla 3.2: Resultados con diferentes combinaciones de NAM y BLC<sup>2</sup>P

### 3.5.3. Análisis de la Autoadaptación de BLC<sup>2</sup>P

En este capítulo hemos presentado el BLC<sup>2</sup>P como una técnica autoadaptable de BLC (ver Sección 3.2). Además, hemos explicado que la autoadaptación puede obtenerse gracias a la combinación de operadores de cruce con capacidad de autoadaptación [BD01] y al proceso de competición integrado en BLC<sup>2</sup>P.

En esta sección tratamos de determinar si la autoadaptación de BLC<sup>2</sup>P realmente permite producir un refinamiento efectivo sobre las soluciones. En particular, comparamos nuestra propuesta, AMCR – BLC<sup>2</sup>P, con un algoritmo similar que usa una BLC que simplemente genera  $n_h \times n_{it}$  hijos a partir de la pareja de padres inicial, y devuelve los dos mejores individuos de entre estos padres y sus hijos (lo llamaremos AMCR de Múltiples Descendientes o AMCR-MD). Durante cada aplicación de la BL de AMCR-MD se generarán 9 hijos ya que AMCR – BLC<sup>2</sup>P utiliza  $n_h = 3$  y  $n_{it} = 3$ . Para realizar una comparación adecuada, no se utilizará el mecanismo adaptable para  $p_{BL}$ , sino un valor fijo para  $p_{BL}$  ( $p_{BL}=0.0625$ ). La Tabla 3.3 recoge los resultados. Al igual que en el apartado anterior, se ha aplicado un *t-test* (a nivel 0.05 de significación) para comprobar

si existen diferencias significativas en los valores de la medida de eficacia  $A$  obtenidos por estos algoritmos. Notaremos, al igual que antes, la dirección de las diferencias significativas de la siguiente forma:

- El signo más (+): la medida  $A$  de  $AMCR - BLC^2P$  es mejor que la de  $AMCR - MD$ .
- El signo menos (-):  $AMCR - MD$  mejora la eficacia  $A$  de  $AMCR - BLC^2P$ .
- Sin signo: no existen diferencias significativas.

Algoritmo	$f_{Esf}$		$f_{Ros}$	
	A	B	A	B
AMCR-MD	8.9e-020+	2.6e-021	1.5e+001+	7.9e+000
AMCR- $BLC^2P$	6.0e-040	3.0e-042	7.7e+000	2.0e-002
Algoritmo	$f_{Sch}$		$f_{Ras}$	
	A	B	A	B
AMCR-MD	2.4e+001+	4.4e-001	1.6e+000	3.0e-011
AMCR- $BLC^2P$	1.3e-003	1.3e-005	1.1e+000	28.0 %
Algoritmo	$f_{Gri}$		$P_{sle}$	
	A	B	A	B
AMCR-MD	1.3e-002	22.0 %	1.3e+001-	1.3e+000
AMCR- $BLC^2P$	1.5e-002	34.0 %	3.3e+001	3.5e+000
Algoritmo	$P_{Cheb}$		$P_{fms}$	
	A	B	A	B
AMCR-MD	9.5e+002+	5.6e+001	3.4e+000-	60.0 %
AMCR- $BLC^2P$	1.8e+002	4.2e+000	6.0e+000	46.0 %

Tabla 3.3: Comparación entre AMCR- $BLC^2P$  y AMCR-MD

Se observa que  $AMCR - BLC^2P$  es más efectivo que  $AMCR - MD$  sobre las funciones unimodales,  $f_{Esf}$ ,  $f_{Ros}$  y  $f_{Sch}$ . Para las multimodales  $f_{Gri}$  y  $f_{Ras}$ , el  $t$ -test indica que no existen diferencias significativas en la efectividad de estos dos algoritmos. Sin embargo, AMCR- $BLC^2P$  obtiene mejores valores para la medida de eficacia  $B$  sobre estas funciones. Para los dos problemas complejos  $P_{sle}$  y  $P_{fms}$ , AMCR- $BLC^2P$  se ve mejorado por AMCR-MD.

Se puede observar que el uso del proceso de competición por parte de  $BLC^2P$  introduce una mayor presión selectiva. En general, gracias a esta característica se

alcanza un adecuado comportamiento sobre muchos problemas. No obstante, para los problemas complejos, la presión selectiva puede acarrear efectos negativos, ya que elimina la capacidad de BLC<sup>2</sup>P para producir saltos adecuados sobre el espacio de búsqueda para descubrir áreas nuevas más prometedoras.

### 3.5.4. Estudio del Mecanismo Adaptable para $p_{BL}$

Existen, al menos, dos caminos para estudiar la operación llevada a cabo por un mecanismo adaptable para AGs [Spe95]. La primera es desde el punto de vista de la efectividad, mediante el empleo de funciones de prueba. El segundo es bastante diferente, ya que ignora la efectividad y se concentra más en el propio mecanismo adaptable, es decir, en su capacidad para ajustar la configuración del AG para adaptarse a las particularidades del problema a resolver. Considerando estos dos aspectos, es razonable investigar la medida en la que el comportamiento adaptativo es responsable de la mejora obtenida.

En esta sección, acometemos el estudio del mecanismo adaptable para determinar  $p_{BL}$ , desde el punto de vista de la propia adaptación. En particular, estamos interesados en determinar si ajusta la proporción  $\frac{L}{G}$  del AMCR propuesto de acuerdo a las particularidades del problema a resolver, permitiendo la obtención de una mejora sobre su operación. Para ello, hemos introducido la Figura 3.4 que muestra para cada problema la media de la ratio  $\frac{L}{G}$  mantenida por el AMCR sobre las 50 ejecuciones (se ha considerado que  $n_{it} = 3$  y  $n_h = 3$ ).

Podemos ver que existen diferencias entre los valores de la ratio  $\frac{L}{G}$  mantenidos para los diferentes problemas. Para las funciones unimodales,  $f_{Esf}$ ,  $f_{Ros}$ , y  $f_{Sch}$ , el algoritmo ha mantenido los valores más altos de ratio, mientras que para los problemas complejos,  $P_{Cheb}$  y  $P_{fms}$ , muestra los valores más bajos. De hecho, el valor de ratio para  $f_{Esf}$  casi duplica el valor para  $P_{fms}$ . Estas diferencias en la ratio  $\frac{L}{G}$  son un signo de la habilidad de adaptación del mecanismo presentado para ajustar  $p_{BL}$ . Estos resultados confirman que este método proporciona valores de la ratio  $\frac{L}{G}$  que se ajustan al problema particular a resolver.

Aunque este mecanismo adaptable muestra capacidad de adaptación, debemos comprobar si la adaptación es verdaderamente beneficiosa. Para ello, hemos ejecutado tres instancias de AMCR con las mismas características que nuestra propuesta, aunque asumen un valor fijo para  $p_{BL}$  (0.0625, 0.25 y 0.5, respectivamente). La Tabla 3.4 contiene los resultados, los cuales van a ser comparados con los de la propuesta, mediante un *t-test*, con un valor de 0.05 de significancia.

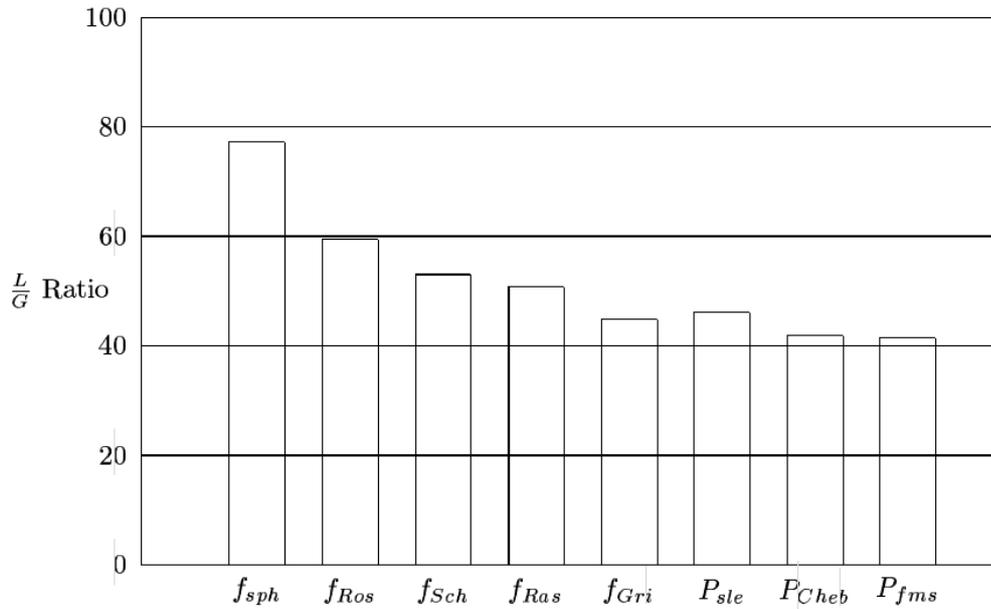


Figura 3.4: Ratio  $\frac{L}{G}$  producida por el AMCR- $BLC^2P$

Al usar  $p_{BL}=0.5$ , se favorece el mantenimiento de altos niveles para la proporción  $\frac{L}{G}$ , obteniéndose resultados muy precisos para las unimodales  $f_{Esf}$ ,  $f_{Ros}$  y  $f_{Sch}$ . Por otro lado, al considerar  $p_{BL}=0.0625$ , se fomentan bajos niveles para la ratio y se alcanzan los resultados más fiables para la función multimodal  $f_{Gri}$  y para los problemas reales. Esto indica que la búsqueda global es adecuada para los problemas complejos y la búsqueda local es más útil sobre los problemas unimodales, lo cual resulta razonable.

Precisamente, la Figura 3.4 revela que, para cada problema de prueba, nuestra propuesta puede alcanzar valores para la ratio  $\frac{L}{G}$  que siguen esta tendencia. Esto explica que pueda alcanzar resultados similares a los obtenidos por la mejor instancia basada en un valor fijo para  $p_{BL}$  (caso de  $f_{Ros}$ ,  $f_{Ras}$ ,  $f_{Gri}$ ,  $P_{Cheb}$  y  $P_{fms}$ ) o, mejor aún, que todas ellas (caso de  $f_{Esf}$ ).

En resumen, este estudio muestra que la capacidad de adaptación del mecanismo presentado para determinar  $p_{BL}$  permite ajustar la proporción  $\frac{L}{G}$  de acuerdo a las particularidades del espacio de búsqueda, permitiendo alcanzar una significativa efectividad sobre problemas con dificultades diferentes.

$p_{BL}$	$f_{Esf}$		$f_{Ros}$	
	A	B	A	B
0.0625	6.0e-040+	3.0e-042	7.7e+000+	2.0e-002
0.25	6.8e-057+	6.3e-061	3.6e+000+	1.4e-004
0.5	7.4e-065+	1.1e-068	2.4e+000	9.5e-004
Propuesta	6.5e-101	1.1e-105	2.2e+000	6.0e-004
$p_{BL}$	$f_{Sch}$		$P_{Ras}$	
	A	B	A	B
0.0625	1.3e-003	1.3e-005	1.1e+000	28.0 %
0.25	1.1e-006	3.7e-009	1.3e+000	40.0 %
0.5	6.6e-008-	1.0e-010	1.4e+000	22.0 %
Propuesta	3.8e-007	4.5e-009	1.4e+000	32.0 %
$p_{BL}$	$f_{Gri}$		$P_{sle}$	
	A	B	A	B
0.0625	1.5e-002	34.0 %	3.3e+001-	3.5e+000
0.25	1.5e-002	28.0 %	8.1e+001+	2.4e+000
0.5	1.7e-002	16.0 %	1.1e+002+	3.2e+000
Propuesta	1.3e-002	30.0 %	5.5e+001	7.9e-001
$p_{BL}$	$P_{Cheb}$		$P_{fms}$	
	A	B	A	B
0.0625	1.8e+002	4.2e+000	6.0e+000	46.0 %
0.25	1.9e+002+	1.1e+001	1.0e+001	22.0 %
0.5	3.5e+002+	4.8e+000	1.2e+001+	20.0 %
Propuesta	1.4e+002	9.2e+000	7.7e+000	40.0 %

Tabla 3.4: Comparación con diferentes valores fijos para  $p_{BL}$  Versus esquema adaptativo del AMCR-BLC<sup>2</sup>P

### 3.6. Estudio Comparativo Frente a otros AEs

En esta Sección, comparamos la eficacia del AMCR propuesto, usando BLC<sup>2</sup>P ( $n_h = 3$  y  $n_{it} = 3$ ) y la probabilidad de aplicación adaptativa, frente a otros modelos propuestos en la literatura. En particular, vamos a hacer dos análisis comparativos. En el primero compararemos nuestra propuesta frente a otros AEs. En el segundo, comparamos la propuesta con otros modelos de AMCRs, un AMCR estacionario estándar y un AMCR basado en el algoritmo CHC.

#### 3.6.1. Comparativa del AMCR-BLC<sup>2</sup>P frente a G3 y AG-FC

En este apartado, vamos a comparar la propuesta planteada con otros modelos más clásicos basados en el cruce.

En particular, vamos a comparar nuestra propuesta con:

- Modelo *Generalized generation gap* (G3) [DAJ02]. El modelo G3 posee el

siguiente modelo iterativo (hasta que no se alcance el criterio de parada):

1. De una población  $P(t)$  se selecciona el mejor padre y  $\mu - 1$  padres aleatoriamente.
2. Se generan  $\lambda$  hijos de los padres seleccionados utilizando un operador de cruce multipadre.
3. Escoger aleatoriamente dos elementos de la población  $P(t)$ .
4. De la subpoblación formada por los dos elementos escogidos y los hijos se seleccionan los dos mejores individuos reemplazando a los individuos escogidos en el paso anterior.

Para este estudio comparativo hemos implementado diferentes instancias G3 que consideran  $\mu = 2$  y emplean  $PBC - \alpha$  ( $\alpha = 1$ ). Se diferencian entre sí en el parámetro  $\lambda$  (1,  $\dots$ , 10, 15, 20 y 50). Denotaremos a estos algoritmos como  $G3 - \lambda$ .

- Competición Familiar (AG-FC) [YK00]. Este modelo incluye una BLC que explora la vecindad de un individuo aplicando de forma consecutiva el cruce con distintas parejas.

Durante el proceso, cada individuo  $I_p$  secuencialmente se convierte en un padre de familia. Con una probabilidad  $p_c$ , al padre de familia y una solución  $I_1$  escogida aleatoriamente de entre los potenciales padres de la población, se les aplica el operadores de cruce. A los descendientes del cruce se les aplica el operador de mutación para generar el conjunto descendiente  $C_1$ . Por cada padre de familia se repite el proceso  $L$  veces, tras lo cual se habrán generado  $L$  soluciones ( $C_1, \dots, C_L$ ) pero sólo la solución  $C_b$  con el mejor valor de adaptabilidad perdura. Posteriormente se aplica una selección de reemplazo para escoger al mejor de entre el padre de familia y su mejor descendiente.

El modelo utilizado ha sido realizado considerando los operadores genéticos originales, y los parámetros asociados elegidos poseen los valores recomendados por Yang y Kao [YK00].

Las Tablas 3.5 y 3.6 muestran el resultado de dicha comparación. Se ha aplicado un *t-test* para identificar si son significativas las diferencias en la medida A entre nuestra propuesta y el resto de algoritmos comparados. El resultado de aplicar el *t-test* es indicado como en los apartados anterior.

Algoritmo	$f_{Esf}$		$f_{Ros}$	
	A	B	A	B
G3-1	9.0e-017	7.9e-103	2.8e+001 +	4.2e+000
G3-2	1.0e-099 +	7.6e-111	1.7e+001 +	6.6e-002
G3-3	6.5e-095 +	1.5e-104	1.1e+001 +	4.0e-005
G3-4	2.2e-085 +	5.6e-098	8.2e+000 +	3.4e-005
G3-5	4.8e-083 +	1.0e-089	1.1e+001 +	3.0e-004
G3-6	4.1e-078 +	1.9e-084	8.4e+000 +	1.6e-003
G3-7	2.1e-073	2.7e-078	9.0e+000 +	1.1e-005
G3-8	4.0e-068	1.1e-072	1.3e+001 +	1.4e-004
G3-9	2.7e-064	2.3e-069	8.5e+000 +	5.9e-003
G3-10	9.9e-062 +	1.3e-065	8.4e+000 +	9.1e-005
G3-15	3.3e-049 +	6.4e-053	1.2e+001 +	2.4e-003
G3-20	2.8e-041 +	7.1e-044	1.7e+001 +	5.0e-003
G3-50	1.2e-021 +	4.6e-024	2.1e+001 +	1.7e-002
AG-FC	1.5e-006 +	3.7e-007	2.3e+001 +	2.1e+001
AMCR-BLC <sup>2</sup> P	6.5e-101	1.1e-105	2.2e+000	6.0e-004
Algoritmo	$f_{Sch}$		$f_{Ras}$	
	A	B	A	B
G3-1	8.3e+002 +	5.3e+001	7.4e+001 +	3.2e+001
G3-2	1.3e+002 +	6.8e-003	6.9e+001 +	3.0e+001
G3-3	9.3e+001 +	4.3e-005	6.9e+001 +	3.5e+001
G3-4	2.9e+001 +	1.8e-005	6.5e+001 +	2.0e+001
G3-5	5.0e+000 +	6.9e-007	6.3e+001 +	4.0e+001
G3-6	9.2e+000	6.9e-007	6.5e+001 +	3.1e+001
G3-7	4.0e+000 +	2.4e-006	6.0e+001 +	2.5e+001
G3-8	3.1e+000	7.2e-007	6.0e+001 +	3.3e+001
G3-9	2.2e+000	1.0e-006	5.0e+001 +	1.4e+001
G3-10	3.8e-001	3.4e-006	6.0e+001 +	3.3e+001
G3-15	8.5e-001	9.0e-007	5.1e+001 +	1.9e+001
G3-20	4.6e-002	1.3e-005	5.2e+001 +	2.5e+001
G3-50	3.5e-001 +	4.0e-003	4.4e+001 +	2.0e+001
AG-FC	1.1e+002 +	4.5e+001	5.5e+000 +	2.2e+001
AMCR-BLC <sup>2</sup> P	3.8e-007	4.5e-009	1.4e+000	32.0 %
Algoritmo	$f_{Gri}$		$P_{sle}$	
	A	B	A	B
G3-1	5.1e-001 +	7.8e-014	3.8e+002 +	3.2e+001
G3-2	2.7e-001	2.0 %	2.2e+002 +	3.1e+001
G3-3	2.3e-001	1.1e-016	2.1e+002 +	3.2e+001
G3-4	4.5e-002 +	8.0 %	1.5e+002 +	1.1e+001
G3-5	3.1e-002 +	8.0 %	1.9e+002 +	7.4e+000
G3-6	3.4e-002 +	4.0 %	1.4e+002 +	1.3e+001
G3-7	3.6e-002 +	16.0 %	1.6e+002 +	4.6e+000
G3-8	2.6e-002 +	16.0 %	1.3e+002 +	8.5e+000
G3-9	2.9e-002 +	4.0 %	1.4e+002 +	6.0e+000
G3-10	2.5e-002 +	8.0 %	1.4e+002 +	7.2e+000
G3-15	1.7e-002	20.0 %	1.2e+002 +	9.5e+000
G3-20	2.1e-002 +	12.0 %	1.0e+002 +	3.4e+000
G3-50	2.0e-002	26.0 %	6.9e+001	3.0e+000
AG-FC	3.5e-004	2.2e-005	2.6e+001 -	7.0e+000
AMCR-BLC <sup>2</sup> P	1.3e-002	30.0 %	5.5e+001	7.9e-001

Tabla 3.5: Comparación de AMCR-BLC<sup>2</sup>P con los algoritmos G3 y el AG-FC, funciones 1-6 (I)

Algoritmo	$f_{Esf}$		$f_{Ros}$	
	A	B	A	B
G3-1	1.9e+003 +	4.7e+001	2.1e+001 +	1.1e+001
G3-2	7.8e+002 +	4.3e+001	1.8e+001 +	4.0 %
G3-3	7.1e+002 +	5.6e+001	1.7e+001 +	4.0 %
G3-4	8.3e+002 +	1.0e+001	1.9e+001 +	5.0e-028
G3-5	8.2e+002 +	1.4e+001	1.7e+001 +	6.0 %
G3-6	4.9e+002 +	4.3e+000	1.7e+001 +	4.0 %
G3-7	5.6e+002 +	6.6e+000	1.6e+001 +	8.0 %
G3-8	4.3e+002 +	4.9e+000	1.6e+001 +	10.0 %
G3-9	5.4e+002 +	3.2e+001	1.7e+001 +	2.0 %
G3-10	6.3e+002 +	1.9e+001	1.6e+001 +	8.0 %
G3-15	4.2e+002 +	2.6e+001	1.5e+001 +	8.0 %
G3-20	3.3e+002 +	1.4e+001	1.5e+001 +	10.0 %
G3-50	2.9e+002 +	3.3e+000	1.2e+001 +	18.0 %
AG-FC	3.9e+002 +	3.9e+001	1.1e+001 +	9.7e-003
AMCR- $BLC^2P$	1.4e+002	9.2e+000	7.7e+000	40.0 %

Tabla 3.6: Comparación de AMCR- $BLC^2P$  con los algoritmos G3 y el AG-FC, funciones 7-8 (II)

Observando los resultados de las Tablas 3.5 y 3.6 podemos extraer las siguientes conclusiones:

- Se observa que conforme aumenta  $\lambda$ , y mayor es el nivel de explotación obtenido durante el cruce, obteniendo mejores resultados.
- Nuestra propuesta es mejor que el 90% de las variantes G3 en 5 de las 8 funciones. Además, en 4 de esas 5, que son funciones con gran complejidad, nuestra propuesta ofrece los mejores resultados.

### 3.6.2. Comparativa del AMCR- $BLC^2P$ frente a otros modelos AMCRs

En este apartado vamos a comparar nuestra propuesta frente a otros AMCRs. Ambos AMCRs emplean Solis Wets [SW81] como método de BL, y se caracterizan en el modelo de AE empleado. Hemos empleado dos modelos, los que ofrecieron los mejores resultados en el capítulo anterior: El clásico y eficiente CHC, y el modelo estacionario clásico (el más parecido al modelo considerado).

- *AMCR basado en un AGE estándar.* Lo constituye un AMCR que sigue un modelo de evolución estacionario y utiliza una estrategia de emparejamiento

aleatoria y una estrategia de reemplazo estándar. A cada nuevo cromosoma generado con PBC- $\alpha$  ( $\alpha = 1$ ) y la mutación BGA se le aplica la técnica de BL Solis-Wets [SW81] con  $p_{BL}=0.0625$ . Hemos ejecutado tres instancias con un número diferente de iteraciones para la BL (100, 1000 y 5000), que hemos llamado AGE-SW-100, AGE-SW-1000 y AGE-SW-5000, respectivamente. Hemos utilizado el AMCR estacionario ya que mantiene el esquema básico utilizado como algoritmo de búsqueda global, y por tanto debe de reflejar de forma más clara las ventajas de nuestras propuestas sobre dicho esquema.

- *AMCRs basados en el algoritmo CHC.* El algoritmo CHC es un referente en la literatura sobre AGs [Esh91]. CHC resulta bastante adecuado para el diseño de AMCRs, ya que incorpora distintas técnicas para promover la diversidad en la población. Hemos usado una implementación de CHC para codificación real (una descripción completa de este algoritmo puede encontrarse en [ECS93]) que aplica el operador PBC- $\alpha$  ( $\alpha = 1$ ). Además, hemos combinado CHC con el procedimiento de BL Solis-Wets, produciendo un AMCR llamado CHC-SW. Cada vez que CHC genera un cromosoma mediante el cruce, la BL se aplica sobre él con  $p_{BL}=0.0625$ . Hemos probado tres instancias de CHC con BL que aplican distintas iteraciones para la BL. Las denotamos CHC-SW-100, CHC-SW-1000 y CHC-SW-5000.

La Tabla 3.7 muestra los resultados. Hemos incluido los resultados del algoritmo CHC (sin BL). Hemos notado nuestra propuesta como AMCR-BLC<sup>2</sup>P. Se ha aplicado una *t-test* para comprobar si las diferencias de AMCR-BLC<sup>2</sup>P con respecto al resto de algoritmos son significativas. La dirección de estas diferencias se representa de igual forma que en la Sección 3.5.3.

Observando la Tabla 3.7, podemos observar que AMCR-BLC<sup>2</sup>P supera a los otros algoritmos. Únicamente, CHC-SW-100 y AGE-SW-5000 mejoran significativamente a AMCR-BLC<sup>2</sup>P sobre tres problemas, y CHC y AGE-SW-1000 sobre dos. En el resto de los casos los resultados de AMCR-BLC<sup>2</sup>P son significativamente mejores.

Por tanto, concluimos que la propuesta se muestra muy competitiva dentro del campo de los AMCRs.

Algoritmo	$f_{Esf}$		$f_{Ros}$	
	A	B	A	B
CHC	5.8e-031 +	3.1e-031	1.9e+001+	1.7e+001
CHC-SW-100	2.1e-014 +	6.7e-015	1.8e+001+	1.6e+001
CHC-SW-1000	9.6e-025 +	4.6e-026	1.5e+001+	7.3e+000
CHC-SW-5000	8.5e-063 +	3.1e-065	1.5e+001+	7.4e+000
AGE-SW-100	3.8e-020 +	5.6e-021	1.0e+001+	1.2e-001
AGE-SW-1000	6.9e-078	1.6e-175	4.5e+000+	1.7e-012
AGE-SW-5000	2.9e-120 -	1.4e-322	4.3e+000+	2.4e-003
AMCR- $BLC^2P$	6.5e-101	1.1e-105	2.2e+000	6.0e-004
Algoritmo	$f_{Sch}$		$f_{Ras}$	
	A	B	A	B
CHC	2.0e-002+	1.5e-003	1.6e+001+	7.0e+000
CHC-SW-100	2.4e+002+	9.4e+001	4.5e+001+	2.9e+001
CHC-SW-1000	1.4e+001+	2.5e+000	6.2e+001+	4.0e+001
CHC-SW-5000	1.2e-001+	1.6e-002	9.4e+001+	5.0e+001
AGE-SW-100	2.9e-007	1.1e-017	7.6e+000+	40.0 %
AGE-SW-1000	5.0e-008 -	6.0e-029	6.8e+001+	2.3e+001
AGE-SW-5000	4.1e-009 -	1.4e-026	1.1e+002+	5.9e+001
AMCR- $BLC^2P$	3.8e-007	4.5e-009	1.4e+000	32.0 %
Algoritmo	$f_{Gri}$		$P_{sle}$	
	A	B	A	B
CHC	6.5e-003 -	42.0 %	3.9e+001	7.7e-001
CHC-SW-100	3.4e-003 -	6.0 %	1.4e+001-	3.7e+000
CHC-SW-1000	2.0e-002 +	4.4e-016	1.5e+002+	3.6e+001
CHC-SW-5000	4.4e-002 +	7.4e-003	3.6e+002+	1.7e+002
AGE-SW-100	2.7e-002	14.0 %	9.1e+000-	6.9e-001
AGE-SW-1000	4.9e-004 -	5.6e-016	1.1e+002+	4.5e+000
AGE-SW-5000	2.9e-003 -	2.6e-015	3.3e+002+	5.4e+001
AMCR- $BLC^2P$	1.3e-002	30.0 %	5.5e+001	7.9e-001
Algoritmo	$P_{Cheb}$		$P_{fms}$	
	A	B	A	B
CHC	3.3e+002+	3.2e+000	1.7e-018-	9.1e-021
CHC-SW-100	1.5e+002	5.3e+000	5.0e+000-	2.1e-015
CHC-SW-1000	6.6e+002+	3.4e+001	1.6e+001+	5.2e-003
CHC-SW-5000	1.4e+003+	4.4e+002	2.0e+001+	1.2e+001
AGE-SW-100	1.0e+002	3.0e+000	1.2e+001+	9.1e+000
AGE-SW-1000	3.6e+002+	8.4e+000	1.5e+001+	8.4e+000
AGE-SW-5000	1.1e+003+	2.2e+002	2.1e+001+	1.4e+001
AMCR- $BLC^2P$	1.4e+002	9.2e+000	7.7e+000	40.0 %

Tabla 3.7: Comparación de AMCR- $BLC^2P$  versus otros AMCRs

### 3.7. Conclusiones

En este capítulo hemos presentado un AMCR. Se ha empleado como AE para la exploración el algoritmo que ofreció los mejores resultados en el capítulo anterior. Se ha propuesto el operador de cruce  $PBC - \alpha$ , especialmente diseñado para su uso dentro de una BL basada en el operador de cruce. Como método de BL, se ha propuesto un método de BLC, denominado  $BLC^2P$ , que hace uso del  $PBC - \alpha$  y que ofrece una alta explotación sin requerir una alta intensidad. Además, hemos propuesto un nuevo modelo de hibridación que determina si aplicar la BL sobre un individuo en función de dicho individuo, para poder explotar en mayor número las soluciones más prometedoras (ajustando  $p_{BL}$  en cada caso). Por último, hemos realizado estudios para medir la influencia de los distintos componentes y comparando la propuesta con otros AMCRs. Las principales conclusiones derivadas de los resultados experimentales son las siguientes:

- La propuesta mejora el resultado obtenido por otros AMCRs aparecidos en la literatura de AMs (considerados en los estudios comparativos de este capítulo).
- Este éxito es parcialmente posible gracias a la combinación de las propiedades del  $NAM$  y a la mejora de soluciones realizadas por  $BLC^2P$ .
- El mecanismo adaptativo  $p_{BL}$  empleado adapta la ratio  $\frac{L}{G}$  generando un comportamiento robusto para las distintas funciones de tests y problemas del conjunto de pruebas.
- El comportamiento autoadaptativo de  $BLC^2P$  ha sido adecuado, en la mayoría de los casos, para obtener niveles de precisión elevados. Sin embargo, muestra ciertas debilidad al afrontar los problemas más complicados.
- Se observa que el empleo de mecanismos de hibridación que utilizan la información del propio individuo a la hora de aplicar la BL permite obtener mejores resultados.

En resumen, los resultados son lo suficientemente prometedores para justificar estudios sobre otros modelos de hibridación adaptativos que hagan uso de la información de los individuos para centrar la BL sobre los individuos más prometedores. En el capítulo siguiente seguimos profundizando en esta cuestión.

## Capítulo 4

# AMCR con Intensidad de BL Adaptativa

Como se ha mostrado hasta ahora, los AMCRs pueden dar muy buenos resultados en optimización real, debido a la combinación de una adecuada *exploración* del espacio de búsqueda, capaz de obtener soluciones de calidad, junto con una alto nivel de *explotación* de las soluciones encontradas, que permita obtener soluciones con un alto grado de precisión.

Los AMCRs se caracterizan en que dichas tareas son realizadas principalmente por un componente distinto: Un AE encargado de realizar la exploración y un método de BL capaz de explotar las soluciones más prometedoras para obtener mayor precisión. Pero dicho buen comportamiento no depende únicamente del AE o del método de BL elegido, sino también de que el esquema de hibridación seguido permita a dichos componentes trabajar de una forma cooperativa [OLZW06, KS05]. Si no hay un adecuado enfoque de hibridación, el uso de métodos de BL no asegura resultados mejores que los del propio algoritmo sin BL [Lan98].

En todo método de hibridación hay dos decisiones importantes involucradas en cómo centrar la BL sobre las regiones más prometedoras:

- *El criterio empleado para seleccionar los individuos a mejorar mediante la BL. ¿Qué individuos de la población deben de ser mejorados mediante la BL, y cómo deben de ser elegidos?*

- *El criterio de parada de la BL.* ¿Cuánto esfuerzo computacional, medido en número de evaluaciones de la función objetivo o *intensidad de la BL* ( $n_{int}$ ), debe de destinarse para cada BL?

En este capítulo, proponemos un AMCR que aplica una BL con una intensidad diferente para cada individuo considerado, es decir, utiliza una intensidad adaptativa. Este proceso implica también un criterio de selección adecuado para elegir los individuos a mejorar, y también para decidir cuánta intensidad debe de aplicarse en cada caso. Esta propuesta es llamada *Algoritmo Memético con Intensidad de BL Adaptativa (Memetic Algorithm with Adaptive Local Search Intensity)*,  $MA^2LSI$ .

Para mostrar los buenos resultados obtenidos por nuestro algoritmo, realizaremos un estudio experimental en el que compararemos los resultados del  $MA^2LSI$  con todos los algoritmos presentados en la Sesión Especial de Optimización Continua del *2005 IEEE Congress of Evolutionary Computation* [SHL<sup>+</sup>05] (CEC'2005).

Este capítulo está estructurado de la siguiente forma: En la Sección 4.1 se detalla la propuesta  $MA^2LSI$ . En la Sección 4.2 se describe el marco experimental empleado en el resto de secciones. En la Sección 4.3 se estudia el comportamiento de nuestra propuesta respecto a sus diferentes parámetros. En la Sección 4.4, para comprobar la bondad de  $MA^2LSI$ , se realiza un estudio comparativo con todos los algoritmos presentados dentro de la sesión especial de optimización con parámetros reales del *2005 IEEE Congress of Evolutionary Computation* [SHL<sup>+</sup>05]. Finalmente, en el apartado 4.5 se indican las conclusiones y posibles trabajos futuros.

## 4.1. Algoritmo Memético con Intensidad de Búsqueda Local Adaptativa

En esta sección se describe nuestra propuesta,  $MA^2LSI$ . Se compone de los siguientes elementos:

- Un AGE [Whi89] responsable de la exploración global. El AGE mantiene los mejores individuos dentro de la población, permitiendo poder aplicar la BL más de una vez sobre el mismo individuo. El AGE empleado es el algoritmo que ofreció los mejores resultados en el análisis del capítulo 2.

- El algoritmo CMA-ES [HO96] para mejorar las soluciones más prometedoras. El CMA-ES es un AE adaptativo para optimización continua, un EDA, que presenta un muy buen comportamiento en funciones unimodales pero no se comporta bien en las multimodales [HK04]. Se describe el CMA-ES en detalle en la Sección 4.1.1.
- Una memoria de BL que almacena asociado a cada individuo el estado final de la aplicación de la BL sobre dicho individuo. Esto hace posible que, cuando se aplica la BL de nuevo a un individuo, el proceso de BL continúe con las mismas condiciones (valores de los parámetros de la BL) con los que el proceso de BL se detuvo la última vez, explotando más esta solución. La memoria de la BL es explicada en detalle en la sección 4.1.2.
- Un modelo de hibridación que adapta la intensidad de la BL en función de la solución a ser mejorada. Ya que el cálculo *a priori* de una adecuada intensidad sería demasiado complejo, este modelo de hibridación aplica la BL varias veces a los mismos individuos (más veces a los más prometedores). Cuando este criterio es empleado conjuntamente con la memoria de la BL, al aplicarse varias veces la BL sobre las soluciones más prometedoras se obtiene un resultado equivalente a aplicar la BL con una intensidad mayor. Esto es explicado con más detalle en la Sección 4.1.3.

Finalmente, en la Sección 4.1.4, se presentan las ventajas de la propuesta presentada.

#### 4.1.1. Componente de Búsqueda Local: CMA-ES

En esta propuesta aplicamos la Adaptación de Matriz de Covarianza, *Covariance Matrix Adaptation* (CMA-ES) como nuestro método de BL [HO01, HK01]. El  $(\mu - \lambda)$  CMA-ES es un algoritmo basado en estimación de distribución (EDA) [LLIB06], que ofrece buenos resultados en optimización real. Posee una alta velocidad de convergencia con la que es capaz de alcanzar soluciones con alto nivel de precisión en muchas funciones unimodales [HMK03], pero en funciones multimodales, no presenta tan buenos resultados como otras técnicas [HK04]. Este comportamiento sugiere que este algoritmo podría ser una BL muy interesante para ser empleada dentro de un AMCR, porque cuando es aplicado como una BL la importancia de su problema con múltiples óptimos es menor (al explorar sólo dentro de una región pequeña el número de óptimos es reducido).

A continuación describimos los principios generales del algoritmo. Puede consultarse [HO01, HK01] para obtener una descripción más detallada.

El  $(\mu, \lambda)$  CMA-ES es un algoritmo que emplea una función de distribución de probabilidad para generar  $\mu$  soluciones, y posteriormente emplea las  $\lambda$  mejores para refinar la propia función de distribución de probabilidad para producir mejores soluciones, en un proceso iterativo. Finalmente, el algoritmo devuelve la mejor solución encontrada.

La matriz de distribución  $x$  es definida mediante:

- El centro de la distribución, definido mediante el punto en el espacio de búsqueda  $\vec{m}$ .
- Una orientación en el eje de coordenadas, definido mediante la matriz de covarianza  $C$ .
- Una varianza, definido por el parámetro  $\sigma$ .

En cada generación  $(g+1)$  el algoritmo sigue los siguientes pasos:

1.  $\lambda$  descendientes son generados de forma independiente según una distribución normal multi-variable.  

$$\vec{x}_k^{(g+1)} \sim N\left(\vec{m} = \langle \vec{x} \rangle_w^g, (\sigma)^2 C^{(g)}\right) \text{ para } k = 1, \dots, \lambda$$
donde  $N(\vec{m}, C)$  denota un vector aleatorio distribuido según una normal con media  $\vec{m}$  y matriz de covarianza  $C$ .
2. Las  $\lambda$  soluciones generadas son evaluadas y ordenadas en función de su valor para la función objetivo.
3. Se seleccionan los  $\mu$  mejores individuos.
4. Considerando únicamente las  $\mu$  mejores soluciones, se actualizan los parámetros  $\vec{m}$ ,  $\sigma$  y  $C$  para centrar la exploración hacia las región que contiene a estos puntos.

El proceso de actualización de parámetros indicado en el paso 4 es el siguiente: Las  $\mu$  mejores nuevos individuos son recombinados para formar una nueva media

$$\langle \vec{x} \rangle_w^{(g+1)} = \sum_{i=1}^{\mu} w_i \vec{x}_{i:\lambda}^{(g+1)}, \text{ donde } w_i \in \mathfrak{R}, w_i \leq 0 \text{ y } \sum_{i=1}^{\mu} w_i = 1. \text{ En [HO01] y}$$

[HK01] se especifican las ecuaciones para actualizar el resto de parámetros: el tamaño de la distribución  $\sigma$ , y la matriz de covarianza  $C$ .

De esta forma, en cada iteración de este proceso las distintas variables se van adaptando: el centro de la distribución  $\vec{x}$  se desplaza hacia una media ponderada de las mejores soluciones encontradas; la matriz de covarianza  $C$  se adapta a la forma de la distribución de las  $\lambda$  mejores soluciones generadas;  $\sigma$  se adapta en función de la mejora obtenida: si durante la BL las nuevas soluciones mejoran a las anteriores, dicho valor es incrementado, y disminuido en caso contrario (para mejorar la precisión reduciendo el espacio de búsqueda). Este proceso iterativo continúa hasta que se haya alcanzado un número máximo de evaluaciones o intensidad,  $n_{int}$ .

Este método requiere únicamente dos parámetros de entrada (para el resto de parámetros de la búsqueda los autores proponen valores por defecto [HK01]): el centro inicial de la distribución  $\langle \vec{x} \rangle_w^{(0)}$ , y el valor  $\sigma$  (el valor inicial  $C$  es la matriz identidad, para que inicialmente la distribución sea esférica). Para poder aplicar  $(\mu - \lambda)$  CMA-ES como un método de BL, asignamos como media inicial de la distribución el propio individuo a mejorar, y se utiliza como valor inicial de  $\sigma$  la mitad de la distancia al individuo más cercano.

#### 4.1.2. Memoria de la Búsqueda Local

Una característica fundamental del modelo propuesto es el uso de una memoria que almacena el estado de la BL tras aplicarse a un individuo. La memoria de la BL permite centrar la búsqueda hacia los mejores individuos, haciendo posible que sea equivalente el aplicar varias veces ( $n_{num}$ ) la BL a un individuo con intensidad  $n_{int}$  que aplicar la BL una única vez con intensidad  $n_{num} \cdot n_{int}$ . Sin dicha memoria, esta equivalencia no sería posible en métodos de BL que, como CMA-ES, se adapta uno o varios parámetros durante el proceso de búsqueda.

En dicha memoria son almacenados junto con los individuos los parámetros que definen el estado final de aplicar la BL sobre él. Así pues, si el mismo individuo es seleccionado de nuevo para ser mejorado mediante la BL, el método de BL puede continuar bajo las mismas condiciones en las que se interrumpió la anterior vez.

Este componente es crucial. Sin él, el modelo de hibridación no funcionaría tal y como se espera. En la sección 4.3.3 se comparan los resultados de emplear este tipo de memoria frente al mismo algoritmo sin emplearla.

En el caso de CMA-ES, los parámetros almacenados son los valores actuales de la matriz de covarianza ( $C$ ), la media de la distribución ( $\vec{m}$ ), el tamaño ( $\sigma$ ) y las distintas variables que influyen en la adaptación de las anteriores. Así pues, únicamente la primera vez se calculan los únicos parámetros necesarios:  $\sigma$  como la mitad de la distancia al individuo más cercano, y el centro inicial de la distribución como la solución a mejorar. El resto de veces todos los parámetros de la BL son obtenidos a partir de los valores finales de la ejecución anterior (a excepción del centro de la distribución, que siempre se actualiza como la solución a mejorar).

### 4.1.3. Modelo de Hibridación

El modelo de hibridación propuesto en este capítulo ha sido diseñado para poder aplicar el método de mejora (BL) con mayor intensidad a los individuos más prometedores, manteniendo al mismo tiempo la aplicación de BL con poca intensidad a soluciones poco prometedoras.

En nuestro modelo mantenemos un valor fijo y definido *a priori* para la ratio de esfuerzo de BL ( $ratio_{BL}$ ), definido como la ratio entre el número total de soluciones evaluadas durante las distintas aplicaciones de la BL, y el número total de evaluaciones. Mantener fija esta ratio nos permite cambiar la intensidad inicial de la BL sin tener que especificar el número de individuos a mejorar (ya que es automáticamente calculado).

El esquema general del método de hibridación es el siguiente:

1. Generar aleatoriamente una población inicial.
2. Aplicar el AGE para evolucionar la población durante  $n_{frec}$  evaluaciones. El parámetro  $n_{frec}$  es el número de evaluaciones del AGE entre dos aplicaciones consecutivas de la BL. El valor  $n_{frec}$  es automáticamente calculado para mantener constante el  $ratio_{BL}$ , mediante la ecuación 4.1.

$$n_{frec} = n_{int} \frac{1 - ratio_{BL}}{ratio_{BL}}. \quad (4.1)$$

Donde  $n_{int}$  es la intensidad de la BL y  $ratio_{BL} = \text{Número de Evaluaciones en la BL} / \text{Número Total de Evaluaciones}$ .

3. Seleccionar de la población global un conjunto de individuos. Únicamente los individuos en este grupo puede ser empleado por la BL.

Un individuo pertenecerá a este grupo si verifica una de las siguientes condiciones:

- a) No haber sido mejorado previamente mediante la BL.
- b) Se le ha aplicado anteriormente la BL y obtuvo una mejora significativa ( $minMejora_{BL}$ ).

Esta condición evita seguir desperdiciando esfuerzo de BL sobre una solución cuando no se obtiene mejora, o ésta es demasiado pequeña (en el caso extremo, podría deberse a ser un óptimo local).

Se podría limitar el número de soluciones prometedoras, pero no lo haremos para permitir la exploración de regiones distantes.

4. Se selecciona un individuo de dicho conjunto, para lo cual aplicamos una medida que indique lo prometedor que consideramos cada individuo. En esta propuesta, elegimos como la solución más prometedoras aquella que presenta el mejor valor de la función objetivo. Existen dos razones para aplicar este criterio:
  - Al utilizar un conjunto no limitado de soluciones prometedoras, el número de individuos es muy alto, por lo que se hace necesario un criterio que centra la BL sobre las buenas soluciones.
  - Si el AE mejora al mejor individuo actual, la siguiente vez que se aplicase la BL se aplicaría sobre dicha nueva solución prometedora.
5. Si el individuo tiene asociado un estado previo de la BL (valores de los parámetros) en la memoria correspondiente, recuperarlos. En otro caso obtener los parámetros de la BL asignándoles sus valores por defecto.
6. Aplicar la BL sobre el individuo seleccionado empleando los parámetros de la BL obtenidos en el paso anterior, utilizando una intensidad de BL ( $n_{int}$ ) fija.
7. Almacenar en la memoria de la BL el estado final de la BL (valores de los parámetros obtenidos al final del proceso de mejora).
8. Ir al paso 2.

#### 4.1.4. Ventajas del Modelo de Hibridación

El modelo de hibridación presenta varias ventajas sobre los modelos de hibridación más clásicos:

- Centra los esfuerzos de la BL sobre las regiones más prometedoras empleando una intensidad de BL adaptada en cada caso al individuo, no aumentando el número de BL sobre dicha región, con el coste en diversidad que eso supone. Por tanto, el uso de una intensidad adaptativa permite incrementar la explotación sobre las regiones más prometedoras sin incrementar el riesgo de convergencia prematura.
- Ha sido diseñado para aprovechar las ventajas de los métodos de BL adaptativos, manteniendo una intensidad lo suficientemente alta para adaptar sus parámetros y obtener así un mejor resultado al aplicarse sobre una solución prometedora.
- El principal parámetro de BL requerido (y, por tanto, que debe ajustarse) es el ratio de esfuerzo  $R$ . El algoritmo es poco sensible al parámetro inicial  $n_{int}$ , porque el valor de intensidad que se aplica se va adaptando a cada función, como puede observarse en el apartado 4.3.5.
- Mantiene constante el ratio de esfuerzo invertido en el componente de BL. Este es un parámetro bastante intuitivo.
- Si el AGE obtiene una nueva mejor solución, esta solución será mejorada por la BL tan pronto como sea posible.

## 4.2. Funciones de Evaluación y el Marco Experimental

Para mostrar el rendimiento de nuestra propuesta hemos considerado el uso de las funciones recomendadas en la *Sesión Especial de Optimización Real del 2005 IEEE Congress of Evolutionary Computation* (CEC'2005 en lo sucesivo) [SHL<sup>+</sup>05]. El uso de estas funciones nos permitirá comparar los resultados obtenidos con nuestro  $MA^2LSI$  con todos los algoritmos propuestos en dicho congreso.

Este apartado se estructura de la siguiente forma: Primero, explicamos las características del marco experimental empleado junto con sus características,

como el número de ejecuciones y evaluaciones. Luego, presentamos los algoritmos utilizados en la comparación. Finalmente, describimos las funciones estadísticas y criterios empleados para nuestro estudio.

#### 4.2.1. Funciones de Test

El conjunto completo de funciones de tests se compone de las siguientes funciones:

- Funciones Unimodales (5): Un único óptimo local (y global).
- Funciones Multimodales (20): Múltiples óptimos locales.
  - Funciones Básicas (7):
  - Funciones Expandidas (2): Sus óptimas globales están situados fuera del rango de inicialización.
  - Funciones Híbridas Cumpuestas (11): Compuesta mediante hibridación de 10 de las funciones anteriores.

Puede consultarse el apéndice para obtener una descripción más detallada, o consultar [SHL<sup>+</sup>05], en donde además de la descripción de las funciones hay un enlace a implementaciones de estas funciones.

Los experimentos con estas funciones se han realizado siguiendo las instrucciones indicadas en el artículo de referencia. Las principales características son:

- Cada algoritmo es ejecutado 25 veces por cada función, y se calcula el error medio del mejor individuo en cada ejecución (se conoce el óptimo en todos los casos).
- Cada experimento es realizado con dimensión 10, 30 y 50.
- Una ejecución termina cuando el error obtenido es mejor que un valor umbral ( $1e-8$ ) o se alcanza el número máximo de evaluaciones. Este número máximo depende de la dimensión, es  $10000 * n_{dim}$  (100000 para dimensión 10, 300000 para dimensión 30, y 500000 para dimensión 50).

Hemos realizado los experimentos siguiendo estas indicaciones ya que fueron las seguidas en el *CEC'2005*, con lo que se posibilita comparar los resultados de

nuestra propuesta con los resultados obtenidos con todos los algoritmos propuestos en dicha competición.

#### 4.2.2. Algoritmos del CEC'2005 comparados

Los algoritmos empleados en este estudio son los siguientes (la notación de los algoritmos es la seguida en otros análisis de dichos algoritmos [Han05a]):

- Varios AEs:
  - K-PCX [STD05]: En este AE, en cada iteración, los nuevos individuos son generados mediante el operador de cruce  $PCX - \alpha$ . En cada iteración se ordenan los individuos por su valor en la función objetivo, y crean  $k$  grupos de individuos igualmente distantes en dicha lista. Dentro de cada grupo se cruzan el mejor de dicho grupo, y otro individuo del mismo grupo, elegido aleatoriamente. Al final de cada iteración se muestrean  $r$  individuos elegidos aleatoriamente para ser sustituidos por los nuevos individuos, si los mejoran.
  - SPC-PNX [BSCG05]: Es un AGE que hace uso del operador de cruce centrado en el padre  $PNX$ , usa como mecanismo de selección la selección uniforme y el reemplazo de *crowding* probabilístico.
  - CoEVO [Pos05]: Es un algoritmo que emplea dos poblaciones para evolucionar la población, una es la población actual, y la otra es un histórico de buenas soluciones alcanzadas anteriores, usada para aumentar la presión selectiva.
  - DE [RKP05]: un esquema DE clásico, que emplea únicamente parámetros fijos.
  - EDA [YG05]: un EDA para optimización real.
- Algoritmos que emplean la estrategia de evolución del CMA-ES:
  - L-CMA-ES [AH05b]: Un sistema multiarranque que emplea CMA-ES como método de BL.
  - G-CMA-ES [AH05a]: Éste es, al igual que el anterior, un sistema multiarranque que aplica CMA-ES. Se diferencia de éste en que cada vez que reinicia se incrementa el parámetro  $\mu$  (y  $\lambda$  para mantener la misma proporción) población multiplicándolo por un factor de incremento.

- Un algoritmo híbrido con BL: BLX-GL50 [GML05]: Es un AE que se caracteriza en emplear un operador de cruce centrado en un padre, el operador de cruce  $PBC - \alpha$  del capítulo anterior, y en el operador de selección: A la hora de elegir los padres, se ordenan en función de su valor objetivo. El padre que se va a utilizar como referencia se escoge aleatoriamente de entre las  $n$  mejores soluciones, mientras que el otro padre se elige aleatoriamente de entre los  $m$  mejores individuos de la población. Tras aplicar durante el 50% de evaluaciones el algoritmo con valores  $n$  y  $m$  altos (fase en la que se fomenta la búsqueda por todo el espacio de búsqueda), se aplica en una segunda fase con valores de  $n$  y  $m$  mucho más reducidos (para explotar las mejores soluciones encontradas).
- Varios algoritmos meméticos:
  - BLX-MA [MHL05a]: Es una propuesta nuestra que se centra en aplicar el modelo de  $p_{BL}$  adaptativa propuesta en el capítulo 3.
  - DMS-L-PSO [LS05]: Un sistema de partículas que aplica un reagrupamiento periódico para fomentar diversidad, y aplica cada cierto número de iteraciones el método de BL Quasi-Newton [Sch81] sobre las mejores soluciones de la población (el 25% de la población).
  - L-SaDE [QS05]: Un DE que combina un mecanismo de autoadaptación, junto con la aplicación cada cierto número de aplicaciones, del método de BL Quasi-Newton al 5% de los individuos.

Se puede consultar [Han05a] o los artículos enviados al *CEC'2005* para obtener más detalles sobre los algoritmos.

Se ha aplicado nuestra propuesta,  $MA^2LSI$ , con los siguientes parámetros:

- Un AGE con una población de 60 individuos. Siguiendo las recomendaciones del capítulo 2, utiliza el cruce BLX-0.5, el método de selección NAM ( $n_{NAM} = 3$ ) y la estrategia de reemplazo  $RW$ . Como mutación se ha aplicado el operador genético BGA con un valor de aplicación por cromosoma al 15.0% de los cromosomas.
- CMA-ES como BL.
- Un método de hibridación con  $minMejora_{BL}$  igual al valor umbral de error,  $1e-8$ . El motivo es evitar que la BL intente mejorar la solución más allá del entorno de óptimo buscado.

En este último capítulo no se han aplicado tests paramétricos. La principal razón es que el conjunto de tests elegido no satisface las condiciones necesarias para emplear comparadores paramétricos [GMLH07, Dem06]. Por tanto, se han aplicado únicamente los tests no paramétricos explicados en el Capítulo 2.

De los distintos algoritmos comentados, cuando tengamos varios algoritmos y queramos identificar cuál es mejor y cuáles son significativamente peores que él, primero aplicaremos el test de Iman-Davenport para confirmar que exista al menos una diferencia significativa, y en ese caso aplicaremos los tests *post-hoc*. El test de Wilcoxon será empleado para comparar un algoritmo (nuestra propuesta) con otro algoritmo (ya que comparando dos algoritmos es más potente que los anteriores), con un valor de significancia ( $p$ ) de 0.05 y 0.10.

Al igual que en el Capítulo 2, complementaremos los resultados con el EAc, que mide el error acumulado del total de funciones consideradas. Para obtener mayor información sobre estos tests, puede consultar las secciones 2.3.1.2 y 2.3.1.3.

### 4.3. Análisis de los Componentes de $MA^2LSI$

En este apartado analizamos nuestra propuesta para estudiar su comportamiento cómo influye cada parámetro, para tratar de identificar sus valores más adecuados. Este apartado posee la siguiente estructura:

- En la Sección 4.3.1, se describen los grupos de funciones para los cuales se van a aplicar este análisis.
- En la Sección 4.3.2, se realizan varios experimentos para determinar una adecuada ratio de esfuerzo ( $ratio_{BL}$ ) y una adecuada intensidad ( $n_{int}$ ).
- En la Sección 4.3.3, se comparan los resultados de  $MA^2LSI$  usando la memoria de la BL y sin usarla, para analizar la conveniencia de su uso.
- En la Sección 4.3.4, se analiza si el método empleado para mejorar las soluciones, el  $CMA-ES$ , se está comportando realmente como un método de BL.
- Finalmente, en la Sección 4.3.5, se comprueba si la intensidad final de la BL aplicada a los distintos individuos es realmente diferente, y si se adapta para cada función.

### 4.3.1. Categorías

Cuando se está trabajando con un número alto de funciones es conveniente, a la hora de realizar un estudio detallado, mostrar los resultados de forma separada para cada categoría de funciones. Las categorías sugeridas por los autores del marco experimental [SHL<sup>+</sup>05] son:

- Funciones Unimodales (funciones 1-5). Estas funciones se caracterizan en la existencia de un único óptimo. La existencia de un único óptimo facilita la explotación por parte de las técnicas de optimización local.
- Funciones Multimodales Básicas, y funciones expandidas (funciones 6-12): Se caracterizan en la existencia de varios óptimos. En este tipo de funciones suelen presentar un robusto comportamiento los AEs.
- Funciones Multimodales expandidas (funciones 13-14): Son funciones cuyo óptimo global está situado fuera del rango de inicialización.
- Funciones Compuestas Híbridas (funciones 15-25). Estas funciones se caracterizan en su definición: Cada una ha sido generada mediante composición de 10 funciones seleccionadas del grupo de funciones anterior (1-14). Las funciones de este grupo presentan la mayor dificultad (sólo una, la 15, ha podido ser resuelto por uno de los algoritmos empleados en la comparación).

De entre estas categorías vamos a centrarnos en nuestros estudios en las funciones multimodales (funciones 6-25), por varios motivos:

- Por un lado, al existir únicamente cinco funciones en la primera categoría, unimodales, no es posible la aplicación de estadísticos sobre estas funciones.
- Por el otro, nos interesa principalmente comprobar cómo se comporta nuestra propuesta en multimodales, ya que en el grupo de unimodales son muy semejantes los comportamientos de gran parte de algoritmos.

Sin embargo, eso no implica que se ignoren las unimodales, a la hora de comparar nuestra propuesta con los otros algoritmos sí se va a hacer un estudio más detallado, en el que se analizarán los comportamientos para la totalidad de las funciones de evaluación, así como respecto a las multimodales.

### 4.3.2. Ajustando los Parámetros de la BL: Ratio de Esfuerzo e Intensidad

En nuestra propuesta existe únicamente dos parámetros que hay que definir, y cuyos valores pueden afectar de forma decisiva la búsqueda: La ratio de esfuerzo ( $ratio_{BL}$ ) y el tamaño de salto de la intensidad ( $n_{int}$ ).

En este apartado aplicamos una serie de experimentos para determinar los valores adecuados de  $ratio_{BL}$  y  $n_{int}$ . Aplicamos un análisis combinado de ambos ya que puede existir sinergia entre estos dos parámetros. Este análisis se aplica se forma independiente para cada dimensión: 10, 30 y 50.

Para los parámetros se han considerado los siguientes valores:

- $ratio_{BL}$ : 0.3, 0.5 y 0.7.
- $n_{int}$ : 100, 500, 1000 y 2000.

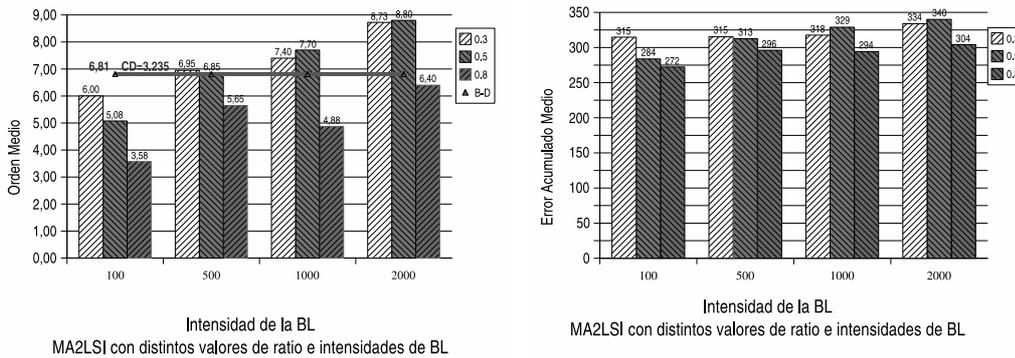
En total, se han aplicado 12 combinaciones diferentes para comparar, denominando a cada algoritmo mediante la notación  $MA^2LSI(ratio_{BL}, n_{int})$ .

Primero aplicamos el test de Iman-Davenport para determinar si el cambiar los parámetros supone una diferencia significativa:

Dimensión	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
10	4.42	1.83	R
30	1.21	1.83	A
50	0.84	1.83	A

Tabla 4.1: Test de Iman-Davenport de  $MA^2LSI(ratio_{BL}, n_{int})$  para cada dimensión, funciones multimodales

Los valores de Iman-Davenport, indicados en la Tabla 4.1 muestran que aunque existen diferencias significativas para dimensión 10, dicha importancia disminuye conforme aumenta la dimensión. A continuación, vamos a hacer un estudio por cada dimensión. Como el test de Iman-Davenport no detecta diferencia significativa para las dimensiones 30 y 50, mostraremos en esos casos únicamente los órdenes medios y los errores acumulados.



(a) Test de Bonferroni-Dunn

(b) Error Acumulado Medio

Figura 4.1: Comparación de las distintas combinaciones  $MA^2LSI(ratio_{BL}, n_{int})$  en multimodales, dimensión 10

**Estudio de ratio e intensidad con dimensión 10.** La Figura 4.1 muestra el resultado de aplicar el test de Bonferroni-Dunn y el EAc para las multimodales. De estas figuras extraemos las siguientes conclusiones:

- Cuanto mayor es la ratio, más veces se aplica la BL, mejores son los resultados.
- Cuanto menor es la intensidad inicial, mejores resultados se obtienen.
- La mejor combinación obtenida es  $MA^2LSI(0,8, 100)$  (tanto en orden medio como en error acumulado), por lo que se recomienda los valores  $ratio_{BL} = 0,8$  y  $n_{int} = 100$  para dimensión 10.

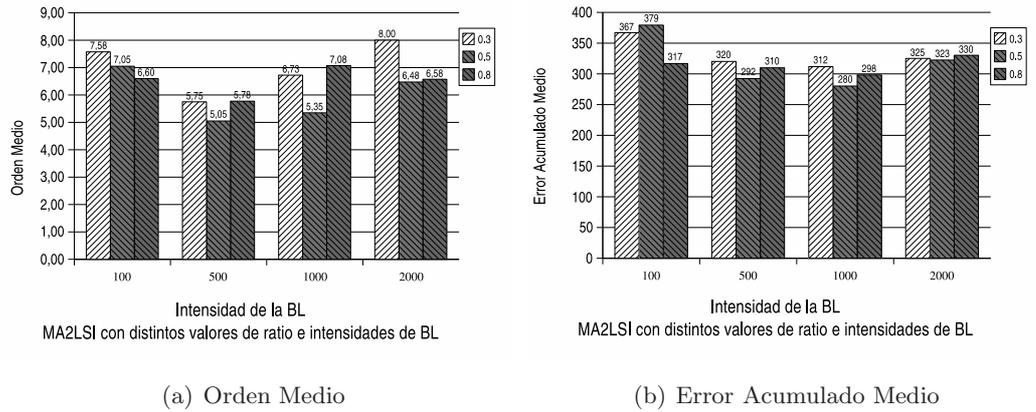


Figura 4.2: Comparación de las distintas combinaciones  $MA^2LSI(ratio_{BL}, n_{int})$  en multimodales, dimensión 30

**Estudio de ratio e intensidad con dimensión 30.** La Figura 4.2 muestra el orden medio y el error acumulado. De dicha figura se pueden extraer las siguientes conclusiones:

- Se obtienen mejores resultados con una intensidad de BL mayor que 100, ya que el incrementar la intensidad de 100 a 500 supone una importante mejora, aunque con mayor intensidad los resultados empiezan a empeorar.
- La intensidad de 500 presenta los mejores resultados, además de mantenerse más robusto frente a cambios de Ratio.
- La variante  $MA^2LSI(0.5, 500)$  se presenta como la más adecuada, por lo que la mejor combinación obtenida es  $ratio_{BL}=0.5$  y  $n_{int}=500$ .

**Estudio de ratio e intensidad con dimensión 50.** La Figura 4.3 muestra el orden medio y el error acumulado. De dicha figura se puede observar las siguientes conclusiones:

- Se obtienen peores resultados con baja ratio (0.3), especialmente en el orden medio.

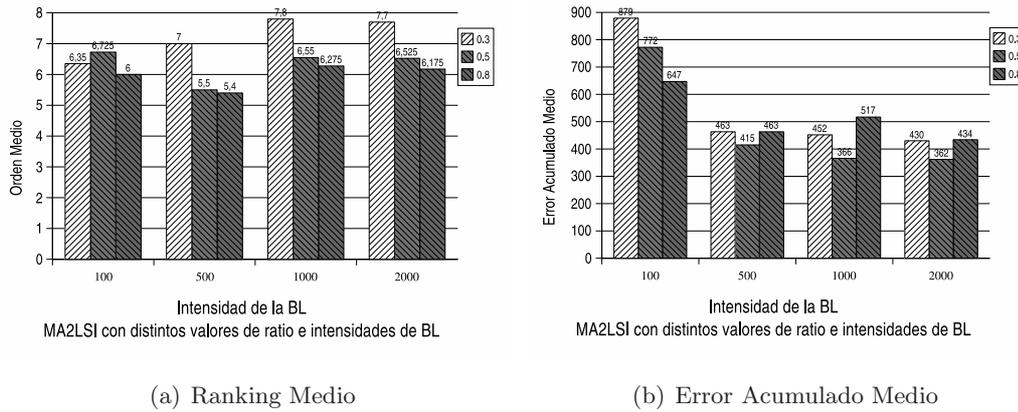


Figura 4.3: Comparación de las distintas combinaciones  $MA^2LSI(ratio_{BL}, n_{int})$  en multimodales, dimensión 50

- Una intensidad de 100 origina un alto error acumulado, por lo que no es recomendable.
- Aparte de los criterios anteriores, no se observa diferencias claras, aunque un ratio de 0.5 parece dar mejores resultados (en error acumulado), por lo que consideramos adecuada la combinación  $MA^2LSI(0.5, 500)$  al presentar un buen comportamiento tanto en orden medio como en error acumulado.

**Conclusiones** Observando los resultados, detectamos que los resultados obtenidos son más similares de lo esperado:

- Parte de dicha similitud se debe a que el tamaño de paso de la intensidad posee una importancia relativa, ya que el criterio adaptativo hace que las soluciones más prometedoras reciban mayor BL sin necesidad de cambiar dicho parámetro.
- Considerando el ratio de la BL, existe otra razón: en nuestro modelo, si la ratio de esfuerzo es suficiente el incrementarla no mejora los resultados. Esto es debido a que en nuestro modelo de hibridación se aplica antes la BL sobre las soluciones más prometedoras, por lo que un incremento en la ratio no es tan determinante como los modelos de hibridación que aplican la

BL de forma aleatoria. En nuestro modelo, un incremento supone aplicar la BL sobre más soluciones, pero menos prometedoras, por tanto su influencia es mucho menor que en otros esquemas de hibridación no dirigidos por las propias características de las soluciones consideradas.

La conveniencia de la elección  $(ratio_{BL}, n_{int})$  depende de la dimensión, por lo que aplicaremos en cada dimensión un valor distinto.

Tras comparar las distintas intensidades y ratios de la BL hemos obtenido que las mejores combinaciones son  $ratio_{BL}=0.8$  y  $n_{int}=100$  para dimensión 10, y  $ratio_{BL}=0.5$  para el resto. Las intensidades adecuadas son  $n_{int} = 500$  y  $n_{int} = 1000$  para dimensión 30 y 50, respectivamente.

### 4.3.3. Influencia de la Memoria de Búsqueda Local

En este apartado estudiamos si mantener en una memoria los valores finales de los parámetros de la BL (y por tanto, una intensidad adaptativa) realmente consigue mejorar los resultados, tal y como fue sugerido en el apartado 4.1.2.

Para comprobarlo, comparamos los resultados de  $MA^2LSI$  (*Con Memoria* en la figura) con una variación que no recupera los parámetros de la memoria, sino que aplica repetidamente sus valores por definen (*Sin Memoria*).

La Tabla 4.2 muestra el resultado de aplicar el test Wilcoxon con ambos algoritmos. En dicha tabla se ha especificado el valor de referencia para el valores de significancia ( $p$ ) de 0.05. La última columna indica si se rechaza o se acepta la hipótesis nula para el margen de error correspondiente (si se acepta ambos algoritmos son equivalentes y si se rechaza existe una diferencia significativa).

Dimensión	MA <sup>2</sup> LSI Elegido	$R^+$	$R^-$	Referencia	Aceptada o Rechazada
Funciones multimodales (6-25)					
10	(0.8, 100)	128	82 (T)	52	A
30	(0.5, 500)	150.5	59.5 (T)	52	A
50	(0.5, 1000)	149	61 (T)	52	A
Todas las funciones (1-25)					
10	(0.8, 100)	216	109 (T)	89	A
30	(0.5, 500)	246	79 (T)	89	R
50	(0.5, 1000)	252	73 (T)	89	R

Tabla 4.2: Test de Wilcoxon Sin Memoria Versus Con Memoria

Este test identifica una mejora respecto a la versión sin memoria.

- El uso de memoria supone una mejora significativa respecto a no usarlo.
- Observando que es mayor la mejora del total de funciones respecto al grupo de multimodales, se puede afirmar que el uso de la memoria es especialmente crucial en las funciones en las que la BL pueda mejorar las soluciones con facilidad, como en las unimodales.
- Cuando la dimensión se incrementa la diferencia entre los algoritmos también se incrementa. Sin embargo, esta tendencia disminuye conforme mayor es la dimensionalidad.

#### 4.3.4. Análisis del Comportamiento de CMA-ES como Búsqueda Local

En nuestra propuesta hemos empleado el algoritmo CMA-ES como nuestro método de mejora. Sin embargo, como CMA-ES es definido como un algoritmo exploratorio, no hay garantía de que se comporte realmente como una BL. Si CMA-ES no se comportase como un método de BL, los buenos resultados podrían ser debidos únicamente a sus buenas características y no a la elección del individuo elegido, es decir, a nuestro proceso de exploración global.

Para comprobarlo, vamos a limitar la exploración CMA-ES en un rango  $[\min_i, \max_i]$  alrededor del punto inicial. El objetivo es asegurar que la solución

resultante de aplicar CMA-ES nunca acabe más cercana a una solución diferente del punto inicial (si una solución es elegida como punto inicial, la BL debería concentrarse sobre ella). Luego, estos dos algoritmos (con y sin esta restricción) son comparados para determinar si CMA-ES se comporta como un método de BL.

Para limitar el dominio de exploración se ha empleado el propio mecanismo de CMA-ES para limitar el dominio de exploración. Para cada dimensión  $i$  se definen los límites superiores e inferiores de dicha región límite  $(min_i, max_i)$  alrededor del punto inicial  $I_{BL}$  mediante el siguiente proceso:

1. Primero, se divide a la población en dos grupos: Los individuos cuyo valor en la dimensión  $i$  es mayor que  $I_{BLi}$ ,  $PMax_i$ , y los individuos cuyo valor  $i$ -ésimo valor es menor o igual que  $I_{BLi}$ ,  $PMin_i$ .
2. Para cada grupo definido en el paso anterior, calculamos  $min_i$  y  $max_i$  mediante las ecuaciones 4.2 y 4.3.

$$max_i = \min(I_{Ni}), \quad \forall I_N \in PMax_i \quad (4.2)$$

$$min_i = \max(I_{Lsi}), \quad \forall I_N \in PMin_i \quad (4.3)$$

En la Figura 4.4 se muestra un ejemplo para dimensión 2.

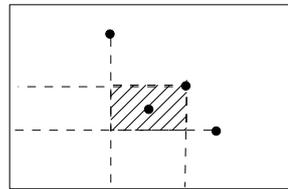


Figura 4.4: Restricción del espacio de búsqueda

Por tanto, en este apartado vamos a comparar el siguiente par de algoritmos:

- En el primero, llamado *Sin restricciones* en la figura, el algoritmo CMA-ES hace la búsqueda considerando todo el espacio de búsqueda.

- En el segundo, llamado *Con restricciones* en la figura, el programa calcula para cada solución a mejorar una región límite tal y como se ha explicado anteriormente, y cada aplicación del CMA-ES se limita dentro de dicha región.

Dimensión	$MA^2LSI$ Elegido	$R^+$	$R^-$	Referencia	Aceptada o Rechazada
10	(0.8, 100)	95 (T)	115	52	A
30	(0.5, 500)	89.5 (T)	120.5	52	A
50	(0.5, 1000)	108.5	101.5 (T)	52	A

Tabla 4.3: Test de Wilcoxon *Con Restricciones* Versus *Sin Restricciones*, funciones multimodales

Como vamos a comparar únicamente dos variantes, vamos a aplicar el test de Wilcoxon. La Tabla 4.3 muestra los resultados de compararlas. Se observa que entre ambas versiones no se han detectado diferencias significativas, por lo que el CMA-ES se comporta efectivamente como una BL en la mayoría de los casos.

#### 4.3.5. Estudio de la Intensidad de la Búsqueda Local, $n_{int}$

En nuestra propuesta se ha propuesto una intensidad adaptable de la BL empleando un tamaño de intensidad  $n_{int}$  fijo. En este apartado comprobamos si la intensidad final aplicada se adapta efectivamente tal y como se mantiene en el apartado 4.3.2.

Para estudiarlo, hemos medido el porcentaje de veces que el método de BL se aplica a una nueva solución. Por motivos de comparación, estas medidas se han aplicado con los mismos valores de BL ( $ratio_{BL} = 0,5$  y  $n_{int} = 1000$ ) para todas las dimensiones, para poder comparar su evolución.

Nuestro objetivo es determinar si la BL realmente se aplica de forma reiterada sobre el mismo individuo, haciendo posible el comportamiento adaptativo de la intensidad.

La Figura 4.5 muestra para cada dimensión y función la ratio de nuevas soluciones mejoradas una o más veces por la BL.

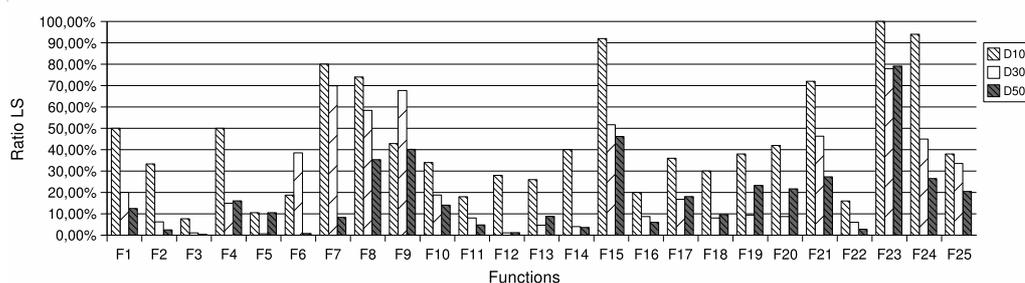


Figura 4.5: Porcentaje de veces que se aplica la BL a una nueva solución (no mejorada anteriormente)

Puede observarse que la ratio de aplicación de BL de una nueva solución depende en gran medida de la función a optimizar. Con una mayor dimensional se incrementa dicha ratio, aumentándose el comportamiento adaptativo: Con dimensión 10, en 6 de las 25 funciones el esfuerzo invertido en nuevas soluciones supone más que el 50%, pero con dimensión 30 este porcentaje se reduce a 3 de las 25 funciones, y únicamente a 1 con dimensión 50. De este comportamiento se observa que, cuando se incrementa la dimensionalidad, la BL se aplica con mayor frecuencia sobre soluciones ya mejoradas, incrementándose así la intensidad efectiva.

La Figura 4.6 muestra para cada función el número de veces que se aplica la BL a soluciones mejoradas previamente, para comprobar que el grado de intensidad final depende de la función considerada. Puede observarse que durante la búsqueda se ha aplicado la BL a muchas soluciones, con diferentes intensidades finales.

En conclusión, un esfuerzo importante del total de aplicaciones de la BL no se emplea para mejorar nuevas soluciones, sino para mejorar soluciones sobre las que se aplicó con anterioridad, demostrando que el modelo de hibridación realmente hace uso de la memoria para adaptar la intensidad de la BL. Además, se observa que la distribución de esfuerzo de BL cambia en función del problema considerado (más para funciones más fáciles, y menos para problemas más complejos), presentando un comportamiento adaptativo.

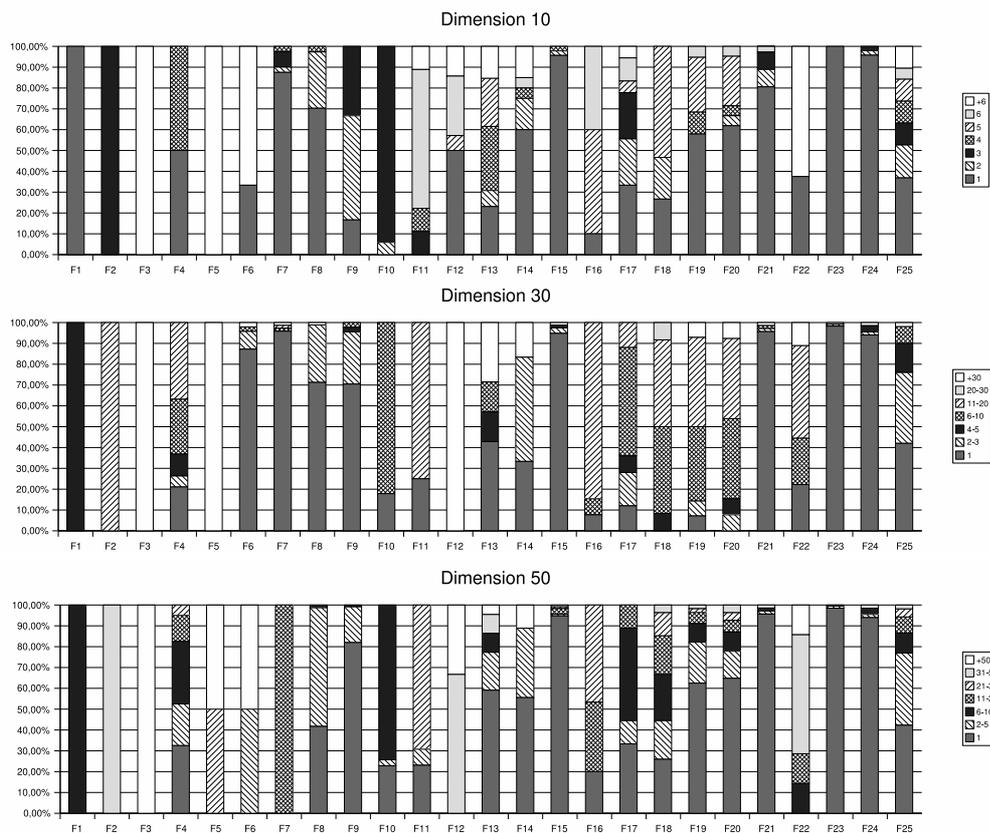


Figura 4.6: Número de veces que se aplica la BL a soluciones mejoradas

#### 4.4. Análisis comparativo con las propuestas de la competición CEC 2005

En este apartado vamos a comparar los resultados obtenidos con nuestra propuesta ( $MA^2LSI$ ) con todos los algoritmos propuestos en la Sesión Especial de Optimización Continua, en el *IEEE Congress on Evolutionary Computation* del 2005 (CEC'2005), enumerados en el apartado 4.2.

Para poder analizar la escalabilidad de nuestra propuesta, dicha comparativa se realizará con las diferentes dimensiones (10, 30 y 50).

#### 4.4.1. $MA^2LSI$ Versus propuestas del CEC'2005, dimensión 10

En este apartado comparamos  $MA^2LSI$  con las distintas propuestas enviadas al CEC'2005 con dimensión 10. Primero aplicaremos los tests estadísticos.

Funciones	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
Todas	4.49	1.83	R
Multimodales	5.74	1.84	R

Tabla 4.4: Test de Iman-Davenport con CEC2005 y  $MA^2LSI(0.8,100)$  para dimensión 10

La Tabla 4.4 muestra el resultado de aplicar el test de Iman-Davenport. Dado que se rechaza para cada grupo de funciones la hipótesis nula, vamos a analizar cada grupo más en detalle (mediante tests *post-hoc*).

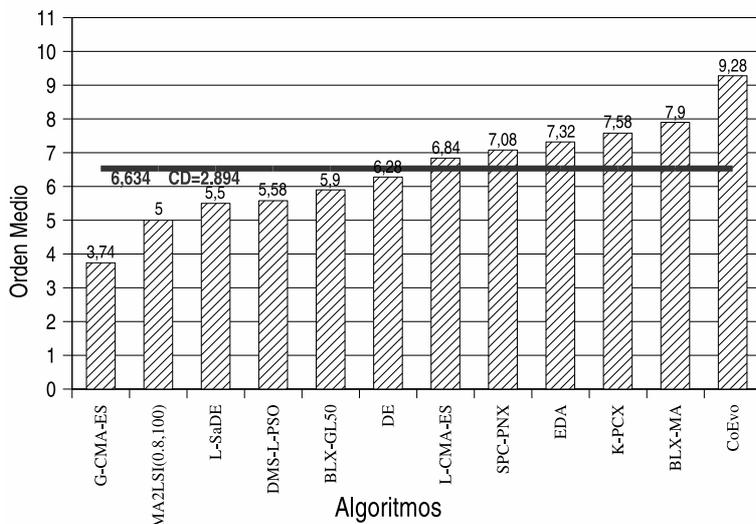
La Figura 4.7 muestra el resultado de aplicar el test de Bonferroni-Dunn tanto sobre el total de funciones como sobre las funciones multimodales. De dicha figura se observa que nuestra propuesta, el  $MA^2LSI$  es la posee el segundo menor error acumulado tanto en global como en multimodales (aunque en este caso esté muy parejo al *SaDE*).

La Figura 4.8 muestra los errores acumulados de cada propuesta. Se observa en este caso que el EAc es muy parecido en ambos casos, y que nuestra propuesta presenta el 4º menor medio.

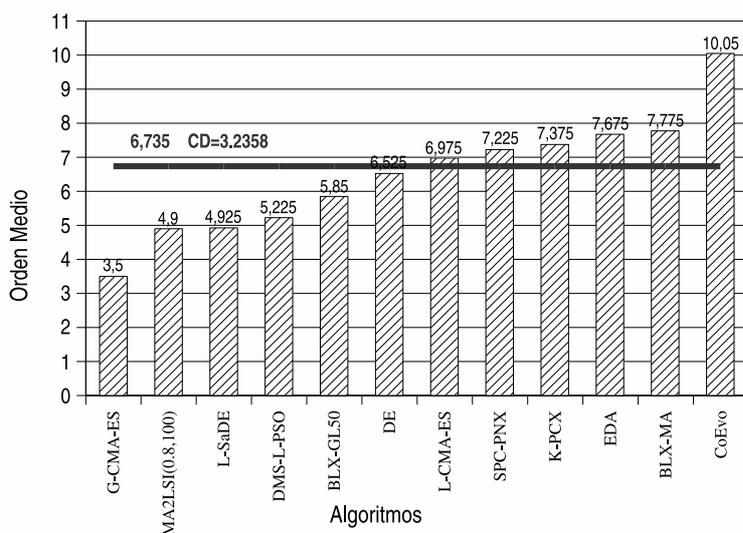
De dichos resultados podemos extraer las siguientes conclusiones:

- Se observa que la propuesta es el segundo mejor algoritmo, sólo después del G-CMA-ES.
- Se mantiene también como uno de los algoritmos con menor error acumulado (el cuarto).

Para comprobar qué algoritmos son mejores o peores que el  $MA^2LSI$ , ya sea de una forma significativa o no, hemos aplicado el test de Wilcoxon de nuestra propuesta con respecto al resto (ver Tabla 4.5).

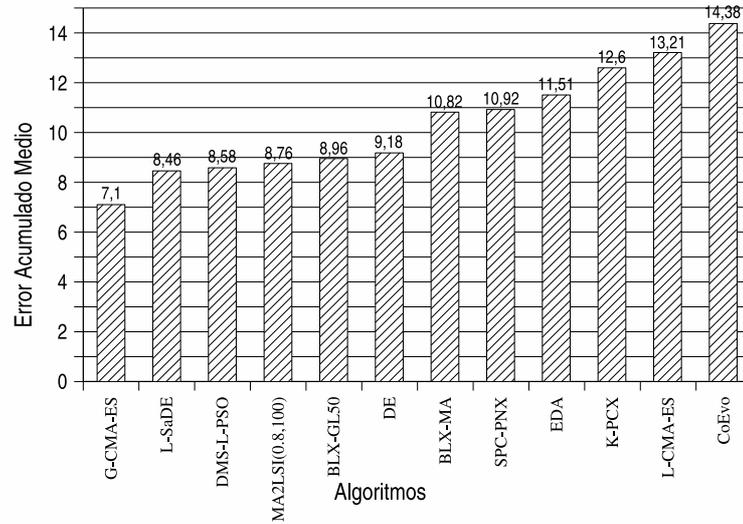


(a) Todas las funciones

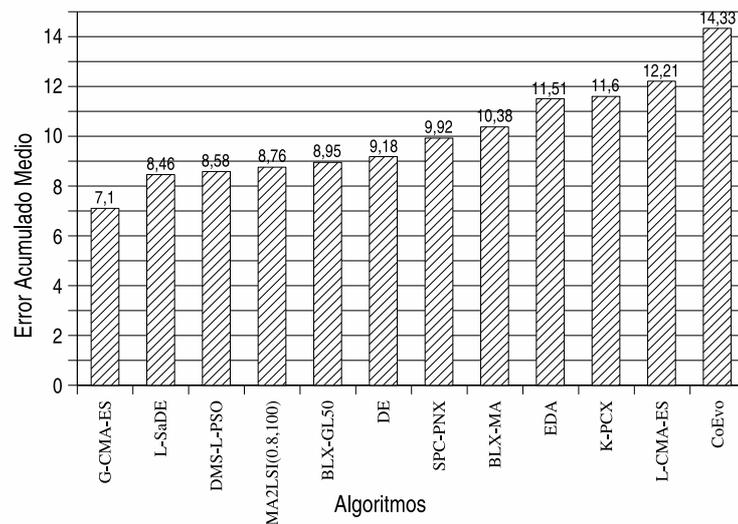


(b) Funciones Multimodales

Figura 4.7: Test de Bonferroni-Dunn sobre CEC'2005 y  $MA^2LSI(0.8, 100)$ , dimensión 10



(a) Todas las funciones



(b) Funciones Multimodales

Figura 4.8: Error Acumulado Medio sobre CEC'2005 y  $MA^2LSI(0.8, 100)$ , dimensión 10

Algoritmo	$R^+$	$R^-$	Referencia	Aceptada o Rechazada
Todas las funciones				
BLX-GL50	188	137 (T)	89	A
BLX-MA	254.5	70.5 (T)	89	R
CoEvo	284	41 (T)	89	R
DE	171	154 (T)	89	A
DMS-L-PSO	160 (T)	165 (T)	89	A
EDA	178	147 (T)	89	A
G-CMA-ES	101.5 (T)	223.5	89	A
K-PCX	278	47 (T)	89	R
L-CMA-ES	210.5	114.5 (T)	89	A
L-SaDE	150 (T)	175	89	A
SPC-PNX	196	129 (T)	89	A
Funciones multimodales				
BLX-GL50	110	100 (T)	52	A
BLX-MA	161	49 (T)	52	R
CoEvo	184	26 (T)	52	R
DE	98 (T)	112	52	A
DMS-L-PSO	115.5	94.5 (T)	52	A
EDA	109	101 (T)	52	A
G-CMA-ES	56.5 (T)	153.5	52	A
K-PCX	180	29.5 (T)	52	R
L-CMA-ES	130.5	79.5 (T)	52	A
L-SaDE	82.5 (T)	127.5	52	A
SPC-PNX	112	98 (T)	52	A

Tabla 4.5: Test de Wilcoxon CEC'2005 Versus MA2LSI(0.8,100), dimensión 10

Grupo de Funciones	Estadísticamente Mejor	Mejor	Peor	Estadísticamente Peor
Todas	3	6	2	0
Multimodales	3	5	3	0

Tabla 4.6: Resumen del test de Wilcoxon CEC'2005 VS MA2LSI(0.8,100), con dimensión 10

La Tabla 4.6 resume los resultados de Wilcoxon, observándose que:

- Nuestra propuesta es estadísticamente mejor que tres algoritmos (BLX-MA, CoEvo, K-PCX), y mejor que otros seis (en total es mejor que el 72 % de los algoritmos).
- Sólo tres algoritmos (G-CMA-ES, L-SaDE, G-CMA-ES y DE) son (según Wilcoxon) mejor que nuestra propuesta. Sin embargo, Wilcoxon no detecta diferencias significativas, por lo que no se puede afirmar que sea peor desde un punto de vista estadístico.

En conclusión, se puede afirmar que nuestra propuesta es, tras el G-CMA-ES, una muy buena opción (segunda según el orden medio). Wilcoxon confirma que no existe ningún otro algoritmo que sea estadísticamente mejor (ya sea en el conjunto total o en las multimodales).

#### 4.4.2. $MA^2LSI$ Versus propuestas del CEC'2005, dimensión 30

En este apartado vamos a comparar las distintas propuestas del CEC'2005 para dimensión 30 con  $MA^2LSI(0.5,500)$ . Nótese que el conjunto de algoritmos comparados es ligeramente menor (son 9 de los 11 empleados con dimensión 10). El motivo es que algunas propuestas presentadas al congreso no presentaron los resultados completos para dimensión 30.

Funciones	Valor Iman-Davenport	$F_F$	Aceptada o Rechazada
Todas	10.12	1.83	R
Multimodales	8.49	1.84	R

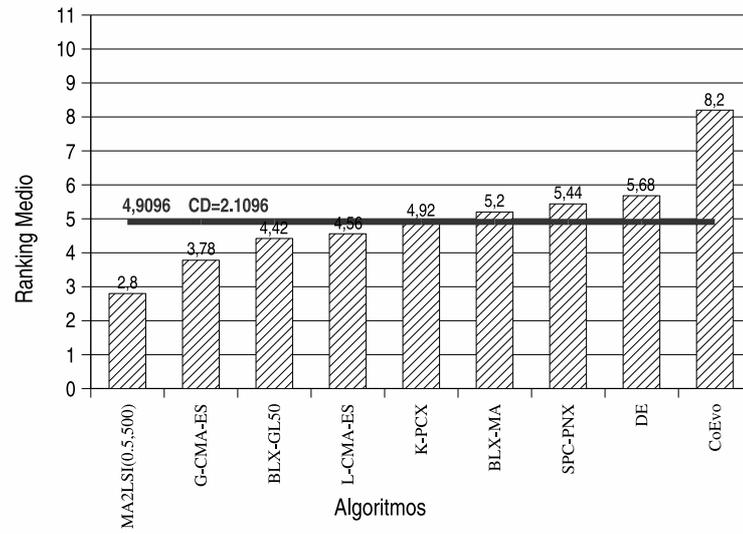
Tabla 4.7: Test de Iman-Davenport con CEC2005 y MA2LSI(0.5,500) para dimensión 30

La Tabla 4.7 muestra el resultado de aplicar el test de Iman-Davenport. Dado que se rechaza con mucho margen para cada grupo de funciones la hipótesis nula, existen diferencias significativas. Vamos a analizar cada grupo más en detalle mediante tests *post-hoc*.

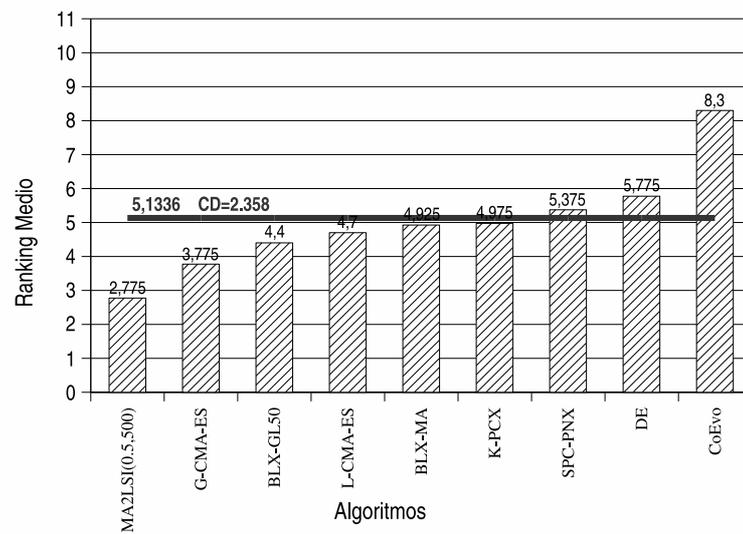
La Figura 4.9 muestra el resultado de comparar todos los algoritmos entre sí, mediante el test de Bonferroni-Dunn. Se puede observar que ahora es nuestra propuesta el que obtiene los mejores resultados.

- Hay diferencias entre ambos grupos en los valores numéricos, pero no en las posiciones relativas de los algoritmos.
- En ambos casos es el que presenta un menor ranking medio.
- Es estadísticamente mejor que el 40 % y el 60 % en los grupos de funciones multimodales y completo, respectivamente. La diferencia no parece deberse a un cambio en el comportamiento. Conforme aumenta el número de funciones consideradas, es más fácil identificar diferencias como significativas.

La Figura 4.10 muestra los errores acumulados de cada propuesta. Se observa que nuestra propuesta presenta el menor EAc, reafirmando la conveniencia de nuestra propuesta.

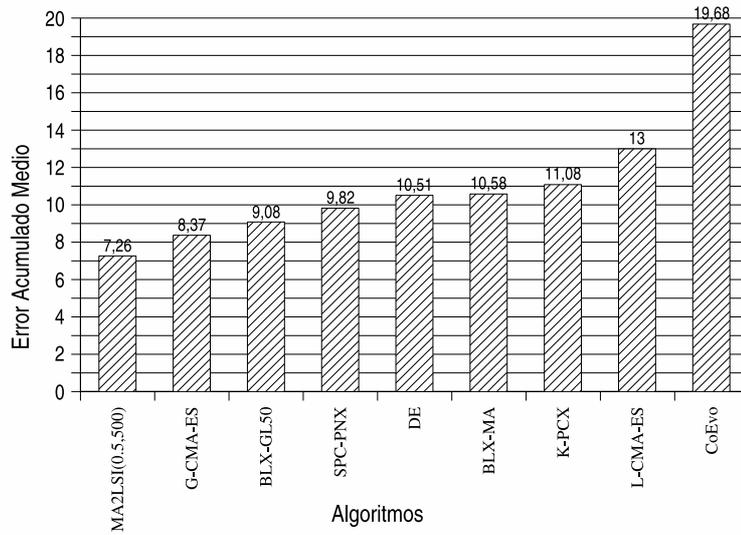


(a) Todas las funciones

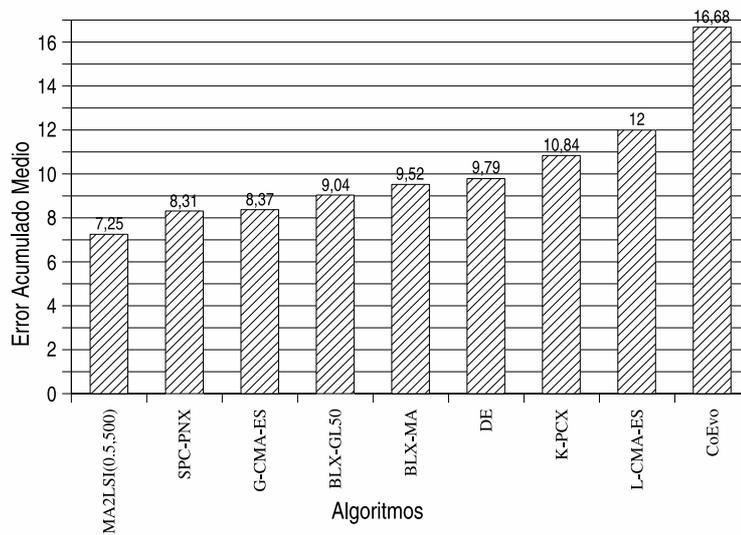


(b) Funciones Multimodales

Figura 4.9: Test de Bonferroni-Dunn sobre CEC'2005 y  $MA^2LSI(0.5, 500)$ , con dimensión 30



(a) Todas las funciones



(b) Funciones Multimodales

Figura 4.10: Error Acumulado Medio sobre CEC'2005 y  $MA^2LSI(0.5, 500)$ , dimensión 30

Algoritmo	Ranking Positivo (R+)	Ranking Negativo (R-)	Referencia	Acceptada o Rechazada
Todas las funciones				
BLX-GL50	264.5	60.5 (T)	89	R
BLX-MA	308.5	16.5 (T)	89	R
CoEvo	325	0 (T)	89	R
DE	309.5	15.5 (T)	89	R
G-CMA-ES	207	118 (T)	89	A
K-PCX	278.5	46.5 (T)	89	R
L-CMA-ES	239	86 (T)	89	R
SPC-PNX	267.5	57.5 (T)	89	R
Funciones multimodales				
BLX-GL50	166	44 (T)	52	R
BLX-MA	198	11.5 (T)	52	R
CoEvo	210	0 (T)	52	R
DE	199.5	10.5 (T)	52	R
G-CMA-ES	139	71 (T)	52	A
K-PCX	174	36 (T)	52	R
L-CMA-ES	165	45 (T)	52	R
SPC-PNX	169.5	40.5 (T)	52	R

Tabla 4.8: test de Wilcoxon CEC'2005 VS MA2LSI(0.5,500), dimensión 30

Para confirmar los algoritmos que son mejores o peores que el nuestro, hemos aplicado el test de Wilcoxon. La Tabla 4.8 muestra los resultados.

Grupo de Funciones	Estadísticamente Mejor	Mejor	Peor	Estadísticamente Peor
Todas	7	1	0	0
Multimodales	7	1	0	0

Tabla 4.9: Resumen del test de Wilcoxon CEC'2005 VS MA2LSI(0.5,500), dimensión 30

La Tabla 4.9 resume los resultados de Wilcoxon, observándose que en comparaciones una a una nuestra propuesta es mejor que todas las del CEC'2005, y estadísticamente mejor que casi todas, (únicamente comparando con G-CMA-ES la diferencia no es estadísticamente significativa).

En conclusión de nuestro análisis en dimensión 30:

- Nuestra propuesta es claramente el mejor algoritmo, y su mejora respecto a casi todos es estadísticamente significativa. Únicamente sobre uno, el CMA-ES, nuestro algoritmo aunque ofrece mejores resultados la mejora no es suficientemente relevantes (desde un punto de vista estadístico).
- El test de Wilcoxon muestra que nuestra propuesta es estadísticamente mejor que casi todos los algoritmos considerados, y mejor que todos ellos. Por tanto, la mejoría en resultados de nuestro modelo es muy importante.
- $MA^2LSI$  posee el menor error acumulado. Esto es muy importante, ya que además de ser el algoritmo que presenta los mejores resultados en más funciones, mantiene un buen comportamiento en el resto.
- La incorporación de un AE sobre el método CMA-ES ofrece un comportamiento más robusto en las funciones multimodales sin que eso suponga reducir su eficiencia en las funciones unimodales.

#### 4.4.3. $MA^2LSI$ Versus propuestas del CEC'2005, dimensión 50

En este apartado vamos a comparar los resultados del  $MA^2LSI$  para dimensión 50. Al igual que pasaba con dimensión 30, el número de algoritmos que ofrece sus resultados para dicha dimensión es muy reducido (únicamente dos: L-CMA-ES y G-CMA-ES). Sin embargo, en nuestro caso son los dos algoritmos que más nos interesan, por usar el mismo método que nosotros utilizamos, CMA-ES. El comparar nuestra propuesta con estas variantes nos permite comprobar las ventajas de mejorar CMA-ES mediante el uso de un AM, frente a los otros enfoques aplicados por estos dos: multiarranque (L-CMA-ES) y un multiarranque con adaptación de parámetros (G-CMA-ES).

Algoritmo	Ranking Positivo (R+)	Ranking Negativo (R-)	Referencia	Aceptada o Rechazada
Todas las funciones				
L-GMA-ES	244	80.5 (T)	89	R
G-CMA-ES	219	106 (T)	89	A
Funciones multimodales				
L-GMA-ES	176	33.5 (T)	52/60	R
G-CMA-ES	154	56 (T)	52/60	R (10%)

Tabla 4.10: Test de Wilcoxon CEC'2005 VS MA2LSI(0.5,1000), dimensión 50

Al ser sólo dos algoritmos, aplicamos directamente el comparador de Wilcoxon. La Tabla 4.10 muestra los resultados.

En este punto es interesante resumir el comportamiento del  $MA^2LSI$  frente a estos dos algoritmos, para cada dimensión.

Dimension	Ranking Positivo (R+)	Ranking Negativo (R-)	Referencia	Aceptada o Rechazada
Todas las funciones				
10	210.5	114.5 (T)	89	A
30	239	86 (T)	89	R
50	244	80.5 (T)	89	R
Funciones multimodales				
10	130.5	79.5 (T)	52	A
30	165	45 (T)	52	R
50	176	33.5 (T)	52	R

Tabla 4.11: Test de Wilcoxon L-CMA-ES Versus MA2LSI

Las Tablas 4.11 y 4.12 muestran los resultados de comparar nuestra propuesta con el L-CMA-ES y el G-CMA-ES, respectivamente. De dichas tablas podemos concluir que:

Dimensión	Ranking	Ranking	Referencia	Aceptada o	
	Positivo (R+)	Negativo (R-)		Rechazada	
	Todas las funciones				
10	101.5 (T)	223.5	89	A	
10	207	118 (T)	89	A	
50	219	106 (T)	89	A	
	Funciones multimodales				
10	56.5 (T)	153.5	52	A	
30	139	71 (T)	52	A	
50	154	56 (T)	52/60	R (10%)	

Tabla 4.12: Test de Wilcoxon G-CMA-ES Versus MA2LSI

- Nuestra propuesta es estadísticamente mejor que L-CMA-ES, es decir, que la vertiente memética es mejor que una variante multiarranque, lo cual confirma que el uso de un AM ofrece una mejora significativa, algo que siempre debe de probarse [CE05]. Es más, la Tabla 4.11 muestra que la mejora es mayor en el grupo más difícil (multimodales) y aumenta conforme aumenta la dimensionalidad (a partir de dimensión 30 la diferencia es significativa).
- Respecto a G-CMA-ES, que se comporta como el mejor algoritmo del congreso (ver [Han05b]), nuestra propuesta es mejor. La Tabla 4.12 muestra que la mejora es mayor en el grupo más difícil (multimodales) y aumenta conforme aumenta la dimensionalidad (con dimensión 50 ya es significativamente mejor con un margen de error de un 10%).

Resumiendo, nuestra propuesta mejora a la propuesta del capítulo anterior, y a todas las propuestas presentadas en el CEC'2005. En este último caso la diferencia es mayor en los casos más complejos (con mayor dimensionalidad y en funciones más complejas). Con dimensión 10 nuestra propuesta es el segundo mejor algoritmo, y estadísticamente equivalente al mejor. Sin embargo, con dimensión 30 nuestra propuesta es siempre el mejor algoritmo, y estadísticamente mejor que casi todas (únicamente respecto al G-CMA-ES la mejora no es suficiente para clasificarse como estadísticamente mejor). Conforme la dificultad aumenta (con dimensión 50 y grupo más complejo) la diferencia se incrementa aún más

y nuestra propuesta es estadísticamente mejor que el G-CMA-ES, demostrando que nuestra propuesta mejora a todas las propuestas del CEC'2005.

## 4.5. Conclusiones

En este último capítulo, hemos aplicado las conclusiones derivadas de los capítulos anteriores (del capítulo 2 el AE adecuado para un AM, el AGE, y del capítulo 3 la conveniencia de un criterio adaptativo de aplicabilidad de la BL) para proponer una propuesta de AMCR. En esta propuesta se emplea una nueva técnica de hibridación que nos permite centrar más la BL (aplicarla con mayor intensidad) sobre las soluciones más prometedoras. Nuestro modelo alcanza esta adaptación de la intensidad de la BL mediante la posibilidad de aplicar de forma reiterada la BL sobre una misma solución, y un sistema de memoria que permite que cada nueva aplicación suponga una continuación de la aplicación anterior. Dada la dificultad de establecer *a priori* una intensidad correcta a cada solución, esta habilidad para adaptar la intensidad de la BL de una forma robusta posee gran valor.

Para comprobar la conveniencia del algoritmo se ha comparado con todos los algoritmos propuestos con la sesión especial de optimización real del *IEEE Congress on Evolutionary Computation*. Las principales conclusiones derivadas de dicha experimentación es:

- Nuestra propuesta no sólo es más robusta, si no que también presenta mejores resultados, especialmente en problemas más complejos (funciones multimodales y con mayor dimensionalidad). Este mejor comportamiento en problemas más difíciles convierte nuestra propuesta en una propuesta de gran interés.
- El éxito de nuestra propuesta es principalmente debida al mecanismo de intensidad adaptativa, que nos permite que se concentre el esfuerzo de la BL hacia las soluciones más prometedoras, aprovechando la mejora alcanzada mediante la aplicación de BL dentro del algoritmo exploratorio.
- Nuestro modelo de hibridación no incrementa el número de parámetros en la búsqueda (respecto a la propuesta no memética), únicamente requiere la ratio de esfuerzo a aplicar en la BL (aunque un valor de 50 % parece bastante robusto).

# Comentarios Finales

Dedicamos esta sección a resumir brevemente los resultados obtenidos y a destacar las principales conclusiones obtenidas en esta memoria. Presentaremos las publicaciones asociadas a esta tesis y comentaremos algunos aspectos sobre trabajos futuros que siguen la línea aquí expuesta, y sobre otras líneas de investigación que se puedan derivar.

## A. Resumen y Conclusiones

En esta memoria hemos presentado distintos AMs que aplican de forma adaptativa la BL para abordar el problema de la optimización continua. El uso de una aplicación adaptativa de la BL se ha realizado con la intención de obtener soluciones precisas (explotación) manteniendo al mismo tiempo una adecuada explotación del espacio de búsqueda. Para ello, se han propuesto dos nuevos métodos de hibridación que, en vez de aplicar la BL sólo sobre el mejor (enfoque demasiado estricto), o aplicar la BL por igual a toda la población (con una probabilidad de aplicación homogénea y baja), toman la decisión de aplicar la Búsqueda Local sobre un individuo teniendo en cuenta las características del propio individuo.

Los siguientes apartados resumen brevemente los resultados obtenidos y presentan algunas conclusiones sobre los mismos.

### A.1 Elección del Algoritmo Explorativo

En esta memoria nos hemos planteado primero estudiar qué tipo de AE es más recomendable para emplearlo como componente de exploración de un AM. Al considerar que la incorporación de un proceso de BL a un AE supone un cambio

en su equilibrio exploración/explotación, no sería fiable elegir un AE con buen comportamiento sin BL, y extrapolar que al incorporarse la BL seguiría siendo una buena elección, sino estudiar su comportamiento tras la incorporación de la BL.

En el Capítulo 2 de esta memoria hemos realizado un estudio comparativo entre distintos AEs, y hemos estudiado cómo evolucionan los resultados conforme se le va incorporando un proceso de BL. De dicho estudio se ha observado que existe mucha diversidad en la mejora de resultados en función del tipo de AE, e incluso del conjunto de operadores considerado. Un algoritmo puede ofrecer por sí muy buenos resultados (CHC), por sus propias características, y puede que que sea difícil de mejorar incorporándole la BL.

Como resultado del capítulo se ha obtenido un AGE que ofrecía el mejor comportamiento, por lo que en los capítulos posteriores se ha empleado dicho algoritmo como algoritmo de exploración, a la hora de proponer nuestros propios modelos meméticos. El AGE elegido posee las siguientes características:

- Población de 60 individuos.
- Método de selección de padres NAM, con  $N_{NAM} = 3$ .
- Operador de cruce BLX-0.5.
- Operador de mutación BGA, al 12.5 % de los cromosomas.
- Estrategia de reemplazo RW.

## A.2 Algoritmo Memético con Búsqueda Local Basada en el Operador de Cruce (AMCR- $BLC^2P$ )

El comportamiento auto-adaptativo de los operadores de cruce, al distribuir los descendientes en función de la distribución de sus padres, es una cualidad adecuada para construir métodos de BL basados en ellos.

En el capítulo 3, hemos propuesto un nuevo método de BL basado en el cruce, denominado  $BLC^2P$ , diseñada para aprovechar esta la capacidad de auto-adaptación de los operadores de cruce para codificación real, constituyendo así un método de BLC *autoadaptable*.

Por otro lado, presentamos un AMCR diseñado para promover altos niveles de diversidad en la población que hace uso de  $BLC^2P$  para mejorar las soluciones

encontradas. Intenta fomentar la *fiabilidad* del proceso de búsqueda asegurando que la BL se centrará sobre diferentes zonas prometedoras a lo largo de la ejecución. Además, el AMCR emplea un mecanismo adaptable que determina, para cada nueva solución, en base a sus valor de la función objetivo, la probabilidad de ser mejorada por dicha BL. De esta forma, este algoritmo ajusta la relación entre búsqueda global y búsqueda local (es decir, la balanza entre exploración y explotación) a las características particulares del problema a resolver.

Las principales conclusiones derivadas de los resultados experimentales son las siguientes:

- Se observa que el uso de  $BLC^2P$  nos permite obtener una alta presión selectiva con un número muy bajo de evaluaciones, lo cual es muy interesante como método de BL, ya que permite una alta explotación pudiendo dedicarse igualmente bastante esfuerzo en la búsqueda global.
- El mecanismo de hibridación propuesto permite que la ratio de esfuerzo (evaluaciones) invertida en la BL se adapte para cada problema considerado, incrementando así la precisión de las mejores soluciones encontradas por la exploración, obteniendo una mejora muy significativa.

En definitiva, se ha propuesto un modelo de AM que combina el uso de una BL propuesta con alta capacidad de explotación junto con un criterio de aplicación adaptativa de la BL que consigue buenos resultados, gracias, en gran parte, a dicha adaptabilidad. Esto nos confirma la importancia de que el mecanismo de hibridación sea adaptativo, aspecto que creemos que puede suponer un tema de estudio de gran interés dentro de los AMs.

### A.3 Propuesta de un AMCR con Intensidad de BL Adaptativa

Tras observar la conveniencia del uso de criterios adaptativos para aplicar la BL, hemos ido un paso más allá: en lugar de aplicar la BL con igual intensidad a todas las soluciones, hemos buscado cómo poder aplicar una intensidad diferente de BL para cada individuo, considerando las características de dicho individuo. El objetivo es poder aplicar la BL con mayor intensidad a las soluciones más prometedoras, lo cual permite introducir como métodos de BL algoritmos adaptativos avanzados que son difíciles de hibridar, ya que requieren un número alto de evaluaciones.

En el Capítulo 4 hemos propuesto un AM capaz de ofrecer una intensidad adaptada a cada solución. Debido a la complejidad de saber *a priori* la intensidad conveniente en cada caso, se ha definido un procedimiento que permite que una misma solución pueda ser elegida más de una vez para ser mejorada por la BL, con una pequeña intensidad fija. Además, se ha definido una memoria de BL que almacena para cada parámetro de dicha búsqueda los valores finales tras cada aplicación. Combinando el uso de la estrategia de hibridación con el empleo de dicha memoria se consigue que sobre las soluciones más prometedoras pueda ser aplicada la BL más veces, continuando en cada caso en el punto en el que se quedó la aplicación anterior, con lo que el efecto de aplicar  $num$  veces la BL con intensidad  $n_{int}$  sea equivalente a una única aplicación de la BL con intensidad  $num \cdot n_{int}$ .

El criterio de aplicación de la BL es sencillo, cada cierto número de evaluaciones se selecciona una solución sobre la que se aplica la BL. Para elegir dicha solución, se considera toda solución que, o no se haya aplicado la BL, o habiéndose aplicado la mejora obtenida sea mayor que un cierto valor umbral. De entre todas éstas, se aplica la BL a aquella con mejor valor de la función objetivo. Así pues, una solución prometedora (con buen valor de la función objetivo) recibirá la atención de la BL mientras no exista otra nueva mejor, y la aplicación de la BL produzca una mejora significativa. En caso contrario, se empieza a aplicar la BL sobre la siguiente mejor.

Hemos aplicado este modelo junto con CMA-ES como método de BL. CMA-ES es un método evolutivo con alta capacidad de explotación de buenas soluciones, y que se perfila como el estado de arte en codificación real para funciones unimodales. Sin embargo, presenta algunos problemas todavía no resueltos en funciones multimodales. Dado que en su uso como BL dentro de un AM, el espacio de búsqueda para el CMA-ES es menor, con lo que se reduce el número de óptimos en su espacio de búsqueda.

Una vez propuesto el AM, denominado  $MA^2LSI$ , hemos realizado los experimentos siguiendo las funciones y los criterios empleados dentro de la Sesión Especial de Optimización Real del *2005 IEEE Congress on Evolutionary Computation*. Esto nos ha permitido comparar nuestra propuesta con todas las propuestas del dicho congreso, propuestas que consideramos representativas de los distintos enfoques para optimización continua. Es de destacar que dentro de dicho congreso los propios autores del CMA-ES presentaron variantes que intentaban mejorar su comportamiento en problemas multimodales, por lo que hemos podido comparar fácilmente nuestro enfoque (uso del CMA-ES original dentro de un AM) con

variaciones multimodales de CMA-ES (G-CMA-ES y L-CMA-ES).

Primero se han valorado la influencia de los factores, obteniendo varias conclusiones interesantes:

- En primer lugar, observamos que la influencia del ratio y de la intensidad inicial es mucho menor que con otros modelos.
- CMA-ES se está comportando realmente como un método de BL.
- El uso de la memoria de BL es fundamental para los buenos resultados.
- La intensidad final aplicada realmente se adapta al individuo, obteniendo una ratio de esfuerzo en BL adaptada al problema

Las comparaciones con los algoritmos del *2005 IEEE Congress on Evolutionary Computation* muestran que nuestra propuesta presenta un buen comportamiento tanto en las funciones unimodales como (y esto es más importante) en las funciones multimodales. Con dimensión 10 es el segundo mejor algoritmo, y al aumentar la complejidad (dimensión del problema) se convierte en el mejor algoritmo de todos.

El resultado final del capítulo se concreta en un nuevo modelo de hibridación sencillo que ofrece una nueva vía de AMs, abriendo al mismo tiempo la gama de métodos de BL que pueden ser empleados en este tipo de algoritmos.

## B. Publicaciones Asociadas a la Tesis

A continuación se presenta un listado de las publicaciones asociadas a la tesis.

- Publicaciones en revistas internacionales:
  1. Herrera F., Lozano M. y Molina D. (2006). Continuous Scatter Search: An Analysis of the Integration of Some Combination Methods and Improvement Strategies. *European Journal of Operational Research* 169:2, 450-476.
  2. Lozano M., Herrera M., Krasnogor N. y Molina D. (2004). Real-Coded Memetic Algorithms with Crossover Hill-Climbing. *Evolutionary Computation* 12:3, 273-302.

- Publicaciones en congresos internacionales:

1. Molina D., Herrera F., Lozano M., García-Martínez C., Sánchez A.M. Sanchez (2005). Técnicas de Diversidad para Algoritmos Meméticos. Actas del Segundo Congreso Mexicano de Computación Evolutiva *COMCEV 2005*, 2005, 29-34.
2. Molina D., Herrera F., Lozano M. (2005). Adaptive Local Search Parameters for Real-Coded Memetic Algorithms. 2005 IEEE Congress on Evolutionary Computation, 2-5 September 2005, Edinburg (Scotland), 888-895.

- Publicaciones en congresos nacionales:

1. Molina D., Herrera F. y Lozano M. (2007). Algoritmo Memético con Intensidad de Búsqueda Local Adaptativa. Aceptado en la *I Jornada sobre Algoritmos Evolutivos y Metaheurísticas, JAEM'2007*, dentro del *II Congreso Nacional de Informática (CEDI 2007)*, Zaragoza, 11-14 de Septiembre, 2007.
2. Molina D., Herrera F. y Lozano M. (2005). Técnicas de Diversidad para Algoritmos Meméticos: un estudio experimental. En las Actas del MAEB'05 (Cuarto congreso español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados), Granada, 39-46, 2005.
3. Lozano M., Herrera F., Krasnogor N., Molina D. (2004) Algoritmos Meméticos con Codificación Real con Técnicas de Ascensión de Colinas Basadas en el Cruce. En Actas del MAEB'04 (Tercer congreso español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados), Córdoba, 254-261, 2004.

## B. Líneas de Investigación Futuras

A continuación, se presentan algunas líneas de trabajo que quedan aún abiertas sobre los temas tratados en estas memorias. Además, consideraremos las extensiones sobre nuestras propuestas que serán objeto de estudios futuros.

### 1. Investigación sobre los componentes de los AMs

Una continuación de este trabajo es el estudio y propuesta de nuevos mecanismos de hibridación, o mejorar los ya presentados considerando no sólo el valor de la función objetivo de las soluciones, sino también otras características.

#### ■ *Definición de nuevos criterios de selección de individuos a mejorar*

En el Capítulo 4 se han propuesto un modelo de hibridación novedoso basado en la idea de la aplicación reiterada de la BL y el uso de una memoria de parámetros de la BL. Sin embargo, un aspecto fundamental para el funcionamiento adecuado es la elección y ordenación de las soluciones prometedoras.

En esta memoria se ha determinado considerar como prometedora cualquier solución sobre la que al aplicarse la BL obtenga una mejora significativa, basándose únicamente en la mejora obtenida por la BL en el paso anterior (si se aplicó) y en su valor para la función objetivo, sin considerar otras características: distancia entre soluciones, mejoras obtenidas en las últimas aplicaciones, número de aplicaciones de BL realizadas, ...

Como se puede intuir, este es un campo en el que existe una gran libertad para definir distintas propuestas, que podrían dar mejores resultados a las propuestas existentes. Como ejemplo:

- Uso de una memoria tabú que impida (durante un tiempo) aplicar dos BLs sobre la misma base de atracción.
- Considerar el número de aplicaciones de la BL, para poder explorar zonas distantes.
- Valorar positivamente no sólo la función objetivo, si no también otro tipo de información del individuo (como la distancia las soluciones ya mejoradas). Así pues, ante soluciones con parecidos valores de la función objetivo, elegiría aquella situada en una región menos explotada.

#### ■ *Diseño de nuevos modelos de hibridación*

Una de las principales conclusiones de esta memoria es la importancia de una adecuada técnica de hibridación. En particular, de la conveniencia de técnicas de hibridación que consideren las propiedades de los propios individuos a la hora de elegir si aplicar la BL y con qué fre-

cuencia. En trabajos futuros se planteará el diseño de nuevos modelos de hibridación.

## 2. *Estudio de nuevos Algoritmos Evolutivos*

La influencia del AE empleado en la Búsqueda Global nos anima a aumentar el estudio comparativo del Capítulo 2 con nuevos AEs como los algoritmos de *Evolución Diferencial* [SP97], Algoritmos Basados en estimación de Distribuciones [LLIB06], Sistemas de Colonias de Hormigas [DS04], o los Sistemas de Partículas [KE95], para poder identificar aquellos más útiles para un AM. Además, sería deseable la obtención de un conjunto de características comunes de los AEs que presenten los mejores resultados, para poder obtener criterios de diseño de los AEs a emplear, que pudiesen seguirse como guía a la hora de definir nuevos AEs específicos para AMs.

## 3. *Paralelismo de los AMs*

El uso de búsqueda locales dentro del proceso de búsqueda puede suponer un alto coste en tiempo. Sin embargo, las distintas mejoras locales podrían realizarse en paralelo, pudiendo mejorar de esa forma el rendimiento del AM, o posibilitando la realización de mayor número de BLs. Se está observando que los AMs son algoritmos potencialmente muy escalables [TLOE06], por lo que es un área de interés el diseñar AMs que faciliten dicho paralelismo, además de utilizar la aplicación paralela de BLs para aumentar el porcentaje de soluciones mejoradas por dichos métodos.

## 4. *Estudio de AMCRs en problemas Multiobjetivos y con Múltiples Soluciones*

El empleo de Algoritmos Meméticos en problemas multiobjetivo se ha consolidado como una propuesta con muchas ventajas [MI95, IN04]. Este hecho, unido a la propuesta del CMA-ES para multimodales [IHR07] hace que trasladar los modelos propuestos a multimodales sea un camino que, aunque complejo, puede ser prometedor. Además, la combinación de una amplia búsqueda con técnicas de mejora local favorece la búsqueda de múltiples óptimos globales distintos.

## 5. *Empleo del modelo en problemas discretos*

El empleo de algoritmos meméticos dentro de los problemas de optimización combinatoria han dado excelentes resultados [Mer00, MK04]. Aunque esta memoria ha estado centrado en los modelos continuos, pensamos que los

modelos de hibridación propuestos son lo suficientemente generales como para poder ser trasladados a problemas combinatorios.



# Apéndice

## Resultados numéricos de $MA^2LSI$

En esta sección presentamos los resultados de  $MA^2LSI$  siguiendo la guía definida en [MHL05b]. El objetivo es permitir a otros autores comparar con nuestra propuesta, siguiendo el espíritu del congreso CEC 2005.

En este apartado presentamos los resultados empíricos obtenidos por el algoritmo propuesto afrontando los 25 problemas indicados [SHL<sup>+</sup>05] con dimensiones 10, 30 y 50. El máximo número de evaluaciones (FES) es de 100000, 300000 y 500000, respectivamente. El margen de error mínimo deseado es de  $1e-8$ , por lo que para valores inferiores el algoritmo se ha interrumpido sin alcanzar el número máximo de evaluaciones.

Presentamos en la Tablas 4.13 y 4.14; 4.15 y 4.16; y 4.17 y 4.18 los resultados empíricos obtenidos por nuestro  $MA^2LSI$  con dimensión 10, 30 y 50, respectivamente. En estas tablas se muestra el error final ( $f(x) - f(x^*)$ ) para cada uno de las 25 ejecuciones. Los valores de las 25 ejecuciones se han ordenado y las tablas presentando las siguientes: 1st (Mejor), 7th, 13th (Mediana), 19th, 25th (Peor), Media y Desviación Estándar (Std). El carácter 'T' después de los errores indican que el error obtenido es inferior a  $1e-8$ . En este caso el algoritmo para antes de que se alcance el máximo FES.

Prob	1	2	3	4
1 <sup>st</sup> (Mejor)	4.245635e-9T	3.378659e-9T	4.568449e-9T	2.947957e-9T
7 <sup>st</sup>	6.670632e-9T	6.569110e-9T	7.342976e-9T	6.779715e-9T
13 <sup>st</sup> (Mediana)	7.863491e-9T	7.458709e-9T	8.681809e-9T	7.963934e-9T
19 <sup>st</sup>	8.595975e-9T	8.617974e-9T	9.038331e-9T	8.362520e-9T
25 <sup>st</sup> (Worst)	9.989776e-9T	9.998018e-9T	9.995460e-9T	9.718292e-9T
Media	7.632359e-9T	7.437015e-9T	8.055370e-9T	7.369247e-9T
Std	1.528119e-9	1.775147e-9	1.542000e-9	1.770384e-9
Prob	5	6	7	8
1 <sup>st</sup> (Mejor)	7.442736e-9T	4.634103e-9T	5.840974e-9T	2.000000e+1
7 <sup>st</sup>	9.392068e-9T	7.179096e-9T	7.858631e-9T	2.000000e+1
13 <sup>st</sup> (Mediana)	1.314623e-8	7.926133e-9T	9.034636e-9T	2.000000e+1
19 <sup>st</sup>	2.007977e-8	8.906341e-9T	7.396048e-3	2.000000e+1
25 <sup>st</sup> (Worst)	3.678126e-7	9.905193e-9T	1.723612e-2	2.000000e+1
Media	4.213918e-8	7.901558e-9T	3.252493e-3	2.000000e+1
Std	8.067998e-8	1.210935e-9	5.163592e-3	0.000000e+0
Prob	9	10	11	12
1 <sup>st</sup> (Mejor)	6.081223e-9T	1.989918e+0	4.202860e-9T	3.574712e-9T
7 <sup>st</sup>	6.986511e-9T	1.989918e+0	4.628899e-3	6.865093e-9T
13 <sup>st</sup> (Mediana)	8.722054e-9T	2.984877e+0	1.315993e-1	9.195787e-9T
19 <sup>st</sup>	9.336645e-9T	2.984877e+0	3.111402e-1	9.787129e-9T
25 <sup>st</sup> (Peor)	9.949591e-1	4.974795e+0	1.947260e+0	1.693622e+3
Media	1.591935e-1	2.785885e+0	3.643913e-1	9.743541e+1
Std	3.722796e-1	8.616598e-1	5.410616e-1	3.616294e+2
Prob	13	14	15	16
1 <sup>st</sup> (Mejor)	3.048120e-1	1.977435e+0	5.704536e-9T	7.683634e+1
7 <sup>st</sup>	3.722357e-1	2.756282e+0	1.481279e+2	9.053571e+1
13 <sup>st</sup> (Mediana)	4.604661e-1	2.908353e+0	2.237845e+2	9.190323e+1
19 <sup>st</sup>	5.619429e-1	3.036174e+0	4.000000e+2	9.305321e+1
25 <sup>st</sup> (Peor)	6.908583e-1	3.297098e+0	4.062585e+2	9.685678e+1
Media	4.674667e-1	2.858167e+0	2.435417e+2	9.086002e+1
Std	1.052171e-1	2.763348e-1	1.373642e+2	5.030551e+0

Tabla 4.13: Valores de Error Alcanzados Con FES=1e5 para Problemas 1-16, dimensión 10

Prob	17	18	19
1 <sup>st</sup> (Mejor)	9.870348e-9	3.000000e+2	3.000000e+2
7 <sup>st</sup>	9.083760e+1	8.000000e+2	3.538308e+2
13 <sup>st</sup> (Mediana)	9.306939e+1	8.000000e+2	8.000000e+2
19 <sup>st</sup>	9.605576e+1	8.227116e+2	8.000000e+2
25 <sup>st</sup> (Peor)	1.028143e+2	9.329263e+2	9.562454e+2
Media	9.017082e+1	7.501701e+2	6.877824e+2
Std	1.917372e+1	2.065190e+2	2.469758e+2
Prob	20	21	22
1 <sup>st</sup> (Mejor)	3.000000e+2	5.000000e+2	3.000000e+2
7 <sup>st</sup>	3.000000e+2	5.000000e+2	7.324169e+2
13 <sup>st</sup> (Mediana)	8.000000e+2	8.000000e+2	7.411986e+2
19 <sup>st</sup>	8.000000e+2	8.000000e+2	7.484203e+2
25 <sup>st</sup> (Peor)	9.363009e+2	1.131399e+3	8.000000e+2
Media	6.996859e+2	6.932560e+2	6.757582e+2
Std	2.463796e+2	1.738016e+2	1.683590e+2
Prob	23	24	25
1 <sup>st</sup> (Mejor)	5.539015e+2	2.000000e+2	4.005056e+2
7 <sup>st</sup>	5.594683e+2	2.000000e+2	4.009459e+2
13 <sup>st</sup> (Mediana)	7.212223e+2	2.000000e+2	4.016255e+2
19 <sup>st</sup>	9.705031e+2	2.000000e+2	4.020245e+2
25 <sup>st</sup> (Peor)	9.705031e+2	5.000000e+2	4.023877e+2
Media	7.576275e+2	2.446447e+2	4.014883e+2
Std	2.020073e+2	9.280048e+1	5.824797e-1

Tabla 4.14: Valores de Error Alcanzados Con FES=1e5 para Problemas 17-25, dimensión 10

Prob	1	2	3	4
1 <sup>st</sup> (Mejor)	5.889603e-9T	7.221104e-9T	7.696826e-9T	9.075677e-9T
7 <sup>st</sup>	7.951712e-9T	7.976325e-9T	9.239045e-9T	1.266156e-5
13 <sup>st</sup> (Mediana)	8.946188e-9T	9.170321e-9T	9.497853e-9T	7.421057e-4
19 <sup>st</sup>	9.453970e-9T	9.745008e-9T	9.759276e-9T	1.628399e-2
25 <sup>st</sup> (Peor)	9.876828e-9T	9.900873e-9T	9.993414e-9T	4.355557e-1
Media	8.608579e-9T	8.897884e-9T	9.426719e-9T	3.204271e-2
Std	1.124308e-9	9.214491e-10	5.156239e-10	9.044482e-2
Prob	5	6	7	8
1 <sup>st</sup> (Mejor)	4.349139e-1	8.896166e-9T	8.102433e-9T	2.000000e+1
7 <sup>st</sup>	1.707470e+1	9.774340e-9T	9.148721e-9T	2.000000e+1
13 <sup>st</sup> (Mediana)	5.278669e+1	9.932251e-9T	9.752881e-9T	2.020271e+1
19 <sup>st</sup>	1.451146e+2	3.864562e+1	7.396040e-3	2.094949e+1
25 <sup>st</sup> (Peor)	1.973359e+2	4.163032e+2	1.231607e-2	2.100621e+1
Media	7.882541e+1	5.244361e+1	2.563049e-3	2.046126e+1
Std	6.953266e+1	1.031235e+2	4.296595e-3	4.761712e-1
Prob	9	10	11	12
1 <sup>st</sup> (Mejor)	1.719513e-10T	1.293447e+1	6.782801e-1	8.824998e-9T
7 <sup>st</sup>	8.119684e-9T	1.790926e+1	3.994936e+0	2.669574e-1
13 <sup>st</sup> (Mediana)	8.925895e-9T	1.890422e+1	4.816318e+0	3.998227e+1
19 <sup>st</sup>	9.531561e-9T	1.989918e+1	5.500045e+0	4.770398e+2
25 <sup>st</sup> (Peor)	9.986366e-9T	2.586893e+1	7.167713e+0	4.289707e+3
Media	8.420466e-9T	1.910321e+1	4.543486e+0	5.024834e+2
Std	2.147610e-09	2.984874e+0	1.593722e+0	9.435907e+2
Prob	13	14	15	16
1 <sup>st</sup> (Mejor)	1.509134e+0	1.206650e+1	3.000000e+2	2.589704e+1
7 <sup>st</sup>	1.700499e+0	1.245655e+1	3.000000e+2	3.571056e+1
13 <sup>st</sup> (Mediana)	2.085893e+0	1.270696e+1	3.000000e+2	3.903249e+1
19 <sup>st</sup>	2.375383e+0	1.291148e+1	3.000000e+2	1.258518e+2
25 <sup>st</sup> (Peor)	2.604679e+0	1.310303e+1	5.000000e+2	5.000000e+2
Media	2.066658e+0	1.266465e+1	3.160000e+2	1.290734e+2
Std	3.674561e-1	3.157949e-1	4.725815e+1	1.721530e+2

Tabla 4.15: Valores de Error Alcanzados Con FES=3e5 para Problemas 1-16, dimensión 30

Prob	17	18	19
1 <sup>st</sup> (Mejor)	2.254108e+1	8.000000e+2	8.000000e+2
7 <sup>st</sup>	3.820403e+1	8.162018e+2	8.163011e+2
13 <sup>st</sup> (Mediana)	4.484458e+1	8.163303e+2	8.163459e+2
19 <sup>st</sup>	4.904445e+1	8.164202e+2	8.164280e+2
25 <sup>st</sup> (Peor)	5.042402e+2	8.167481e+2	8.166720e+2
Media	1.005092e+2	8.150706e+2	8.157084e+2
Std	1.357057e+2	4.539290e+0	3.274701e+0
Prob	20	21	22
1 <sup>st</sup> (Mejor)	8.161473e+2	1.181494e+3	1.204033e+3
7 <sup>st</sup>	8.162189e+2	1.213454e+3	1.273238e+3
13 <sup>st</sup> (Mediana)	8.163306e+2	1.219082e+3	1.297994e+3
19 <sup>st</sup>	8.163985e+2	1.231367e+3	1.315472e+3
25 <sup>st</sup> (Peor)	8.166662e+2	1.239872e+3	1.399870e+3
Media	8.163366e+2	1.219883e+3	1.297771e+3
Std	1.500183e-1	1.414298e+2	3.806754e+0
Prob	23	24	25
1 <sup>st</sup> (Mejor)	1.166853e+3	1.200656e+3	1.310833e+3
7 <sup>st</sup>	1.212431e+3	1.260555e+3	1.435643e+3
13 <sup>st</sup> (Mediana)	1.221830e+3	1.278009e+3	1.461869e+3
19 <sup>st</sup>	1.232530e+3	1.295061e+3	1.474307e+3
25 <sup>st</sup> (Peor)	1.244128e+3	1.306014e+3	1.540526e+3
Media	1.217250e+3	1.271156e+3	1.453590e+3
Std	3.547299e-4	0.000000e+0	2.215331e+0

Tabla 4.16: Valores de Error Alcanzados Con FES=3e5 para Problemas 17-25, dimensión 30

Prob	1	2	3	4
1 <sup>st</sup> (Mejor)	5.889603e-9T	7.221104e-9T	7.696826e-9T	9.075677e-9T
7 <sup>st</sup>	7.951712e-9T	7.976325e-9T	9.239045e-9T	1.266156e-5
13 <sup>st</sup> (Mediana)	8.946188e-9T	9.170321e-9T	9.497853e-9T	7.421057e-4
19 <sup>st</sup>	9.453970e-9T	9.745008e-9T	9.759276e-9T	1.628399e-2
25 <sup>st</sup> (Peor)	9.876828e-9T	9.900873e-9T	9.993414e-9T	4.355557e-1
Media	8.608579e-9T	8.897884e-9T	9.426719e-9T	3.204271e-2
Std	1.124308e-9	9.214491e-10	5.156239e-10	9.044482e-2
Prob	5	6	7	8
1 <sup>st</sup> (Mejor)	4.349139e-1	8.896166e-9T	8.102433e-9T	2.000000e+1
7 <sup>st</sup>	1.707470e+1	9.774340e-9T	9.148721e-9T	2.000000e+1
13 <sup>st</sup> (Mediana)	5.278669e+1	9.932251e-9T	9.752881e-9T	2.020271e+1
19 <sup>st</sup>	1.451146e+2	3.864562e+1	7.396040e-3	2.094949e+1
25 <sup>st</sup> (Peor)	1.973359e+2	4.163032e+2	1.231607e-2	2.100621e+1
Media	7.882541e+1	5.244361e+1	2.563049e-3	2.046126e+1
Std	6.953266e+1	1.031235e+2	4.296595e-3	4.761712e-1
Prob	9	10	11	12
1 <sup>st</sup> (Mejor)	1.719513e-10T	1.293447e+1	6.782801e-1	8.824998e-9T
7 <sup>st</sup>	8.119684e-9T	1.790926e+1	3.994936e+0	2.669574e-1
13 <sup>st</sup> (Mediana)	8.925895e-9T	1.890422e+1	4.816318e+0	3.998227e+1
19 <sup>st</sup>	9.531561e-9T	1.989918e+1	5.500045e+0	4.770398e+2
25 <sup>st</sup> (Peor)	9.986366e-9T	2.586893e+1	7.167713e+0	4.289707e+3
Media	8.420466e-9T	1.910321e+1	4.543486e+0	5.024834e+2
Std	2.147610e-09	2.984874e+0	1.593722e+0	9.435907e+2
Prob	13	14	15	16
1 <sup>st</sup> (Mejor)	1.509134e+0	1.206650e+1	3.000000e+2	2.589704e+1
7 <sup>st</sup>	1.700499e+0	1.245655e+1	3.000000e+2	3.571056e+1
13 <sup>st</sup> (Mediana)	2.085893e+0	1.270696e+1	3.000000e+2	3.903249e+1
19 <sup>st</sup>	2.375383e+0	1.291148e+1	3.000000e+2	1.258518e+2
Std	6.786439e-1	4.489861e-1	7.571801e+1	1.342398e+2

Tabla 4.17: Valores de Error Alcanzados Con FES=5e5 para Problemas 1-16, dimensión 50

Prob	17	18	19
1 <sup>st</sup> (Mejor)	3.703511e+1	8.381925e+2	8.374783e+2
7 <sup>st</sup>	4.451780e+1	8.386623e+2	8.378862e+2
13 <sup>st</sup> (Mediana)	4.778811e+1	8.394637e+2	8.385393e+2
19 <sup>st</sup>	5.407421e+1	8.403701e+2	8.395702e+2
25 <sup>st</sup> (Peor)	4.113900e+2	9.489307e+2	9.437260e+2
Media	1.327202e+2	8.592394e+2	8.465273e+2
Std	1.576019e+2	3.872915e+1	2.762682e+1
Prob	20	21	22
1 <sup>st</sup> (Mejor)	8.376769e+2	5.000000e+2	5.000082e+2
7 <sup>st</sup>	8.378509e+2	5.000000e+2	5.000155e+2
13 <sup>st</sup> (Mediana)	8.385393e+2	5.000000e+2	5.000197e+2
19 <sup>st</sup>	8.390402e+2	5.000000e+2	5.000237e+2
25 <sup>st</sup> (Peor)	8.407959e+2	5.000000e+2	5.000528e+2
Media	8.386488e+2	5.000000e+2	5.000211e+2
Std	8.934384e-1	0.000000e+0	1.006840e-2
Prob	23	24	25
1 <sup>st</sup> (Mejor)	5.391222e+2	2.000000e+2	2.160114e+2
7 <sup>st</sup>	5.391222e+2	2.000000e+2	2.167715e+2
13 <sup>st</sup> (Mediana)	5.391227e+2	2.000000e+2	2.170561e+2
19 <sup>st</sup>	5.391227e+2	2.000000e+2	2.174612e+2
25 <sup>st</sup> (Peor)	1.173849e+3	2.000000e+2	2.183310e+2
Media	5.923993e+2	2.000000e+2	2.170892e+2
Std	1.547789e+2	0.000000e+0	5.718003e-1

Tabla 4.18: Valores de Error Alcanzados Con FES=5e5 para Problemas 17-25, dimensión 50

## Funciones de tests del CEC'2005

El conjunto completo de funciones de tests se compone de las siguientes funciones:

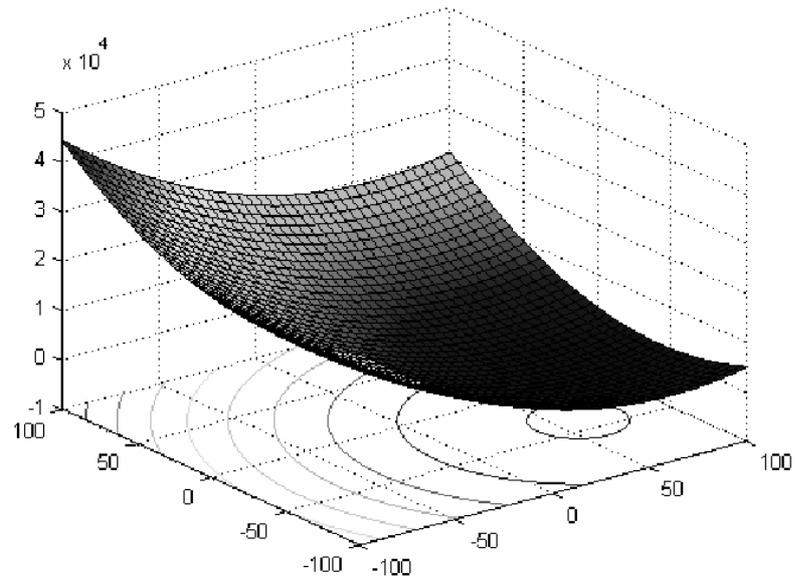
- Funciones Unimodales (5): Un único óptimo local (y global).
  - Función esférica desplazada (*Shifted Sphere Function*).
  - Problema desplazado 1.2 de Schwefel (*Shifted Schwefel's Problem 1.2*).
  - Función elíptica desplazada y altamente condicionada (*Shifted Rotated High Conditioned Elliptic Function*).
  - Problema desplazado 1.2 de Schwefel con ruido (*Shifted Schwefel's Problem 1.2 with Noise in Fitness*).
  - Problema 2.6 de Schwefel con los óptimos situados en la frontera (*Schwefel's Problem 2.6 with Global Optimum on Bounds*).
- Funciones Multimodales (20): Múltiples óptimos locales.
  - Funciones Básicas (7):
    - Función de Rosenbrock desplazada (*Shifted Rosenbrock's Function*).
    - Función de Griewank desplazada y rotada sin límites en la búsqueda (*Shifted Rotated Griewank's Function without Bounds*).
    - Función de Ackley desplazada y rotada con el óptimo global en las fronteras (*Shifted Rotated Ackley's Function with Global Optimum on Bounds*).
    - Función de Rastrigin desplazada (*Shifted Rastrigin's Function*).
    - Función de Rastrigin desplazada y rotada (*Shifted Rotated Rastrigin's Function*).
    - Función de Weierstrass desplazada y rotada (*Shifted Rotated Weierstrass Function*).
    - Problema 2.13 de Schwefel (*Schwefel's Problem 2.13*).
  - Funciones Expandidas (2): Sus óptimas globales están situados fuera del rango de inicialización.
    - Funciones de Griewank y Rosenbrock expandidas y extendidas (*Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)*).

- 
- Función Scaffer expandida, extendida y rotada (*Expanded Rotated Extended Scaffer's F6*).
  - Funciones Híbridas Cumpuestas (11): Compuesta mediante hibridación de 10 de las funciones anteriores.
    - Función híbrida compuesta 1 (*Hybrid Composition Function 1*).
    - Función híbrida compuesta 1 rotada (*Rotated Hybrid Composition Function 1*).
    - Función híbrida compuesta 1 con ruido (*Rotated Hybrid Composition Function 1 with Noise in Fitness*).
    - Función híbrida compuesta 2 (*Rotated Hybrid Composition Function*).
    - Función híbrida compuesta 2 con un estrecho base de atracción sobre el óptimo (*Rotated Hybrid Composition Function 2 with a Narrow Basin for the Global Optimum*).
    - Función híbrida compuesta 2 con óptimos globales en la frontera (*Rotated Hybrid Composition Function 3 with the Global Optimum on the Bounds*).
    - Función híbrida compuesta 3 (*Rotated Hybrid Composition Function 3*).
    - Función híbrida compuesta 3 rotada y altamente condicionada (*Rotated Hybrid Composition Function 3 with High Condition Number Matrix*).
    - Función híbrida compuesta 3 rotada y no-contínua (*Non-Continuous Rotated Hybrid Composition Function 3*).
    - Función híbrida compuesta 4 rotada (*Rotated Hybrid Composition Function 4*).
    - Función híbrida compuesta 4 rotada y sin frontera (*Rotated Hybrid Composition Function 4 without Bounds*).

A continuación detallamos las distintas funciones. En donde  $f(x)$  representa la función de evaluación sobre la solución  $x$ ,  $o$  es el óptimo global, y  $D$  es la dimensión (10, 30 ó 50).

**F1: Función Esférica Desplazada**

$$f(x) = \sum_{i=1}^D z_i^2 + f\_bias, z = x - o, x = [x_1, x_2, \dots, x_D]$$

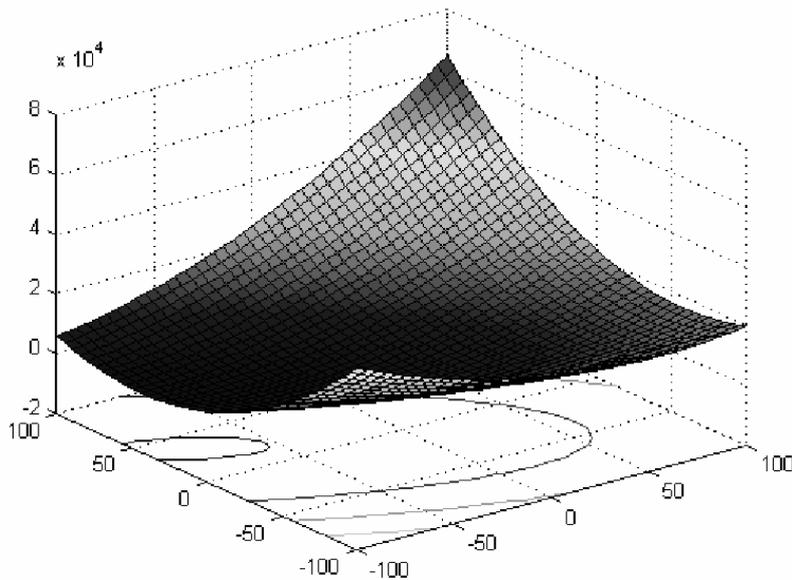


Propiedades:

- Unimodal.
- Desplazada.
- Separable.
- Escalable.
- $x \in [-100, 100]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) = f\_bias(1) = -450$ .

## F2: Problema 1.2 de Schwefel Desplazado

$$f(x) = \sum_{i=1}^D \left( \sum_{j=1}^i z_j^2 \right) + f\_bias, z = x - o, x = [x_1, x_2, \dots, x_D]$$

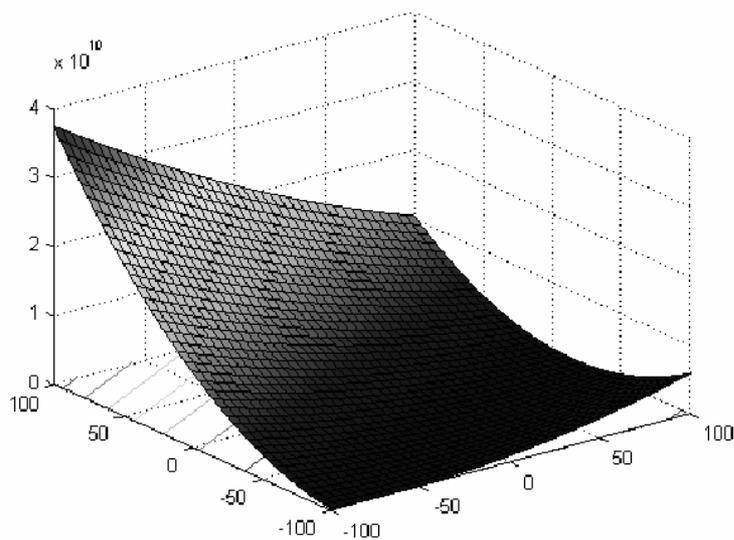


Propiedades:

- Unimodal.
- Desplazada.
- No Separable.
- Escalable.
- $x \in [-100, 100]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) = f\_bias(2) = -450$ .

### F3: Función Elíptica Desplazada y Altamente Condicionada

$$f(x) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} z_i^2 + f\_bias, z = x - o, x = [x_1, x_2, \dots, x_D]$$

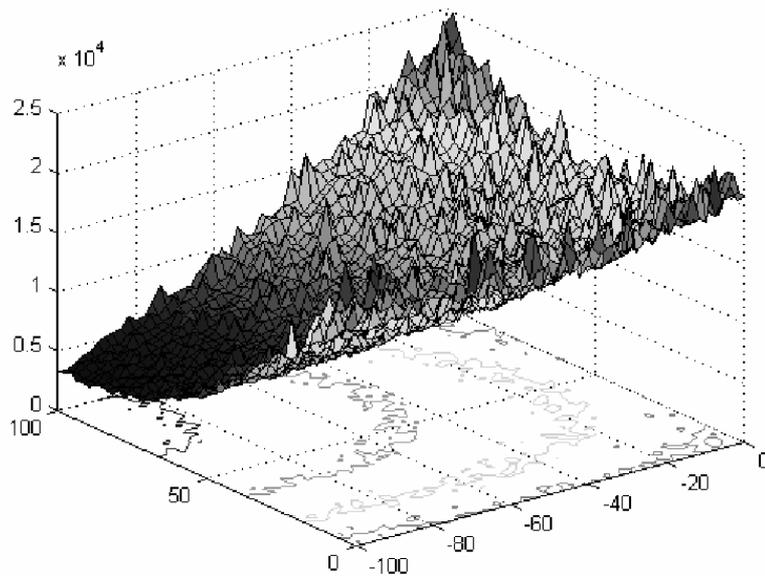


Propiedades:

- Unimodal.
- Desplazada.
- Rotada.
- No Separable.
- Escalable.
- $x \in [-100, 100]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) = f\_bias(3) = -450$ .

#### F4: Problema Desplazado 1.2 de Schwefel con Ruido

$$f(x) = \left( \sum_{i=1}^D \left( \sum_{j=1}^i z_j \right)^2 \right) \cdot (1 + 0,4|N(0, 1)|) + f\_bias, z = x - o, x = [x_1, x_2, \dots, x_D]$$



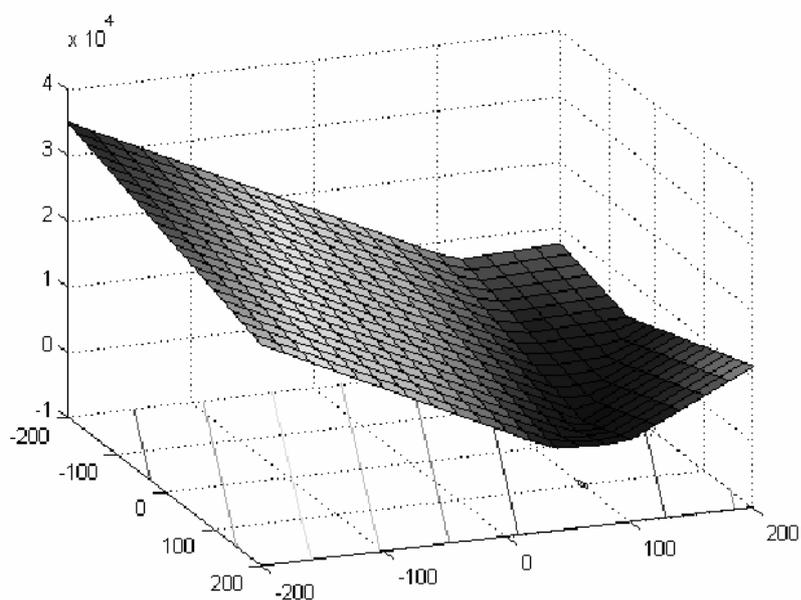
Propiedades:

- Unimodal.
- Desplazada.
- No Separable.
- Escalable.
- Con ruido en la función de evaluación.
- $x \in [-100, 100]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) = f\_bias(4) = -450$ .

### F5: Problema 2.6 de Schwefel con los Óptimos Situados en la Frontera

$$f(x) = \max |A_i x - B_i| + f\_bias, i = 1, \dots, D, x = [x_1, x_2, \dots, x_D]$$

Donde  $A$  es una matriz  $D * D$ ,  $a_{ij}$  son números aleatorios en el rango  $[-500, 500]$ ,  $\det(A) \neq 0$ ,  $A_i$  es la  $i$ -ésima fila de  $A$ .  $B_i = A_i * o$ ,  $o$  es un vector  $D * 1$ ,  $o_i$  son números aleatorios en el rango  $[-100, 100]$ ,  $o_i = -100 \forall i = 1, 2, \dots, \lceil D/4 \rceil$ ,  $o_i = 100 \forall i = \lceil 3D/4 \rceil, \dots, D$ .



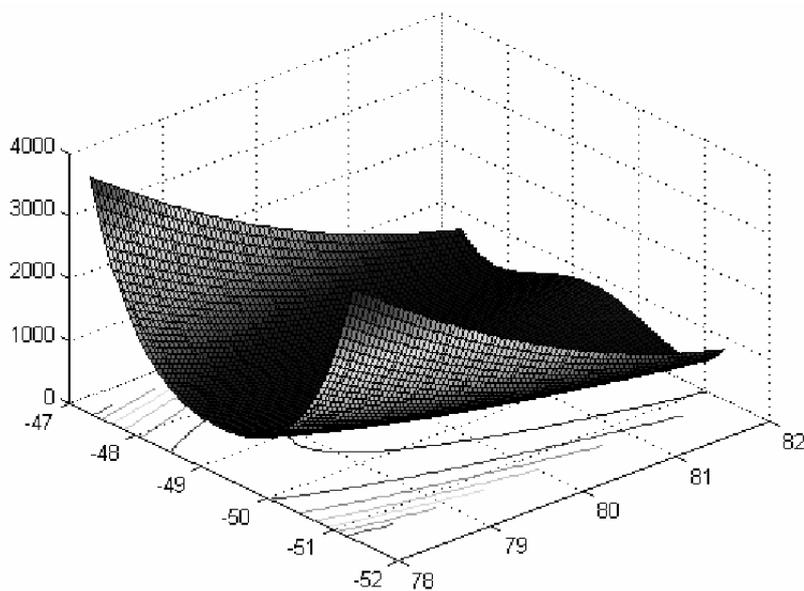
Propiedades:

- Unimodal.
- No Separable.
- Escalable.

- Si el procedimiento de inicialización inicializa a la población en la frontera, la función será resuelta fácilmente.
- $x \in [-100, 100]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) = f\_bias(5) = -510$ .

### F6: Función de Rosenbrock Desplazada

$$f(x) = \sum_{i=1}^D (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f\_bias, z = x - o + 1, x = [x_1, x_2, \dots, x_D]$$



Propiedades:

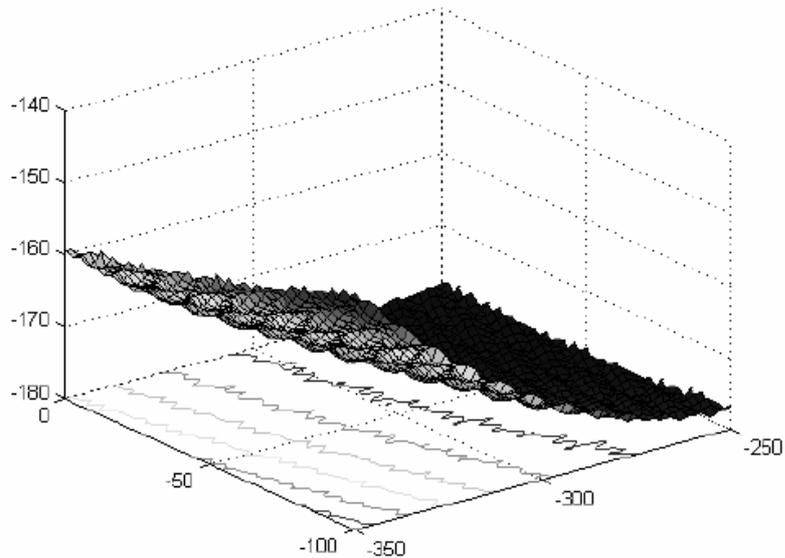
- Multi-modal.
- Desplazada.
- No Separable.

- Escalable.
- Posee un valle desde el óptimo local al óptimo global
- $x \in [-100, 100]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) = f\_bias(6) = 390$ .

### F7: Función de Griewank Desplazada y Rotada sin Límites en la Búsqueda

$$f(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f\_bias, z = (x - o) * M, x = [x_1, x_2, \dots, x_D]$$

Donde  $M = M \cdot (1 + 0,3|N(0, 1)|)$ , donde  $M'$  es una matriz de transformación lineal, con 3 de número de condición.



Propiedades:

- Multi-modal.

- Rotada.
- Desplazada.
- No Separable.
- Escalable.
- Sin frontera.
- Se inicializa la población en  $[0, 600]^D$ , óptimo global  $x^* = o$  fuera de dicho rango,  $f(x^*) = f\_bias(7) = -180$ .

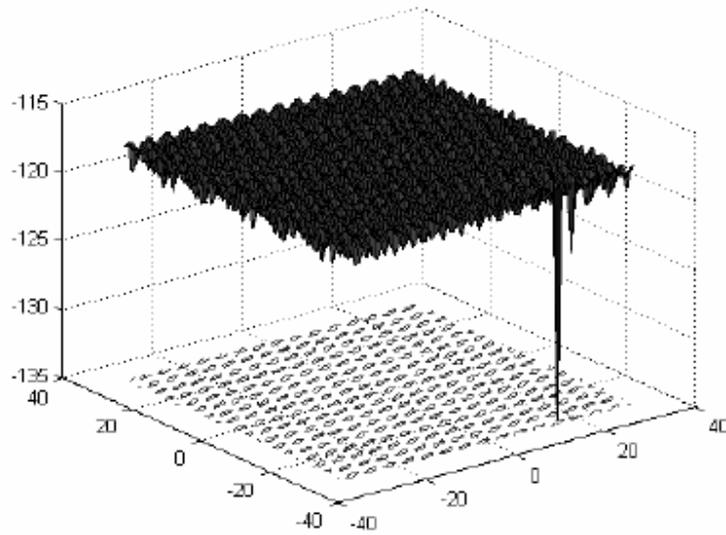
### F8: Función de Ackley Desplazada y Rotada con el Optimo Global en la Frontera

$$f(x) = -20 \cdot \exp\left(-0,2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right) + 20 + e + f\_bias$$

$$z = (x - o) * M, x = [x_1, x_2, \dots, x_D]$$

Donde  $o_{2j-1} = -32o_{2j}$ , y está aleatoriamente distribuidos para  $j = 1, 2, \dots, \lfloor D/2 \rfloor$ .

$M = M \cdot (1 + 0,3|N(0, 1)|)$ , donde  $M'$  es una matriz de transformación lineal, con 100 de número de condición.

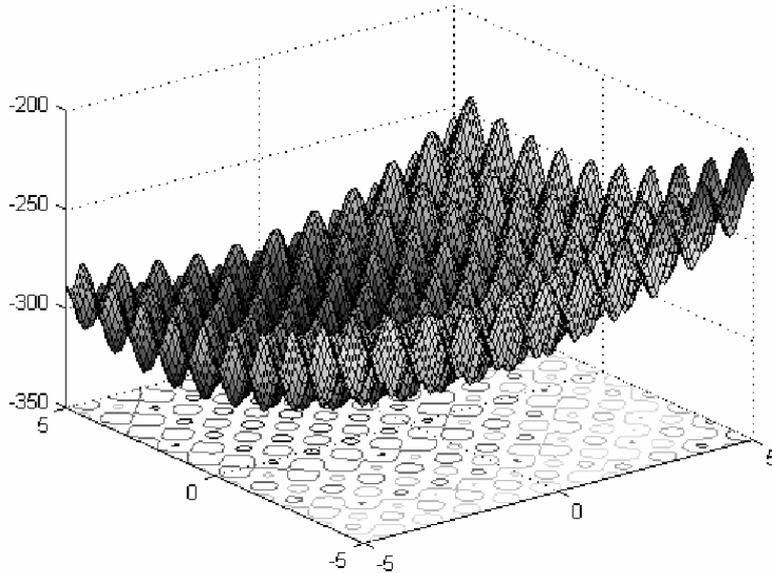


Propiedades:

- Multi-modal.
- Rotada.
- Desplazada.
- No Separable.
- Escalable.
- El número de condicionantes de A se incrementa con el número de variables con  $O(D^2)$ .
- Óptimo global en la frontera.
- Si el procedimiento de inicialización inicializa a la población en la frontera, la función será resuelta fácilmente.
- $x \in [-32, 32]^D$ , óptimo global  $x^* = o$ ,  $f(x^*) = f\_bias(8) = -140$ .

## F9: Función de Rastrigin Desplazada

$$f(x) = \sum_{i=1}^D (z_i^2 - 10\cos(2\pi z_i) + 10) + f\_bias, z = (x - o), x = [x_1, x_2, \dots, x_D]$$



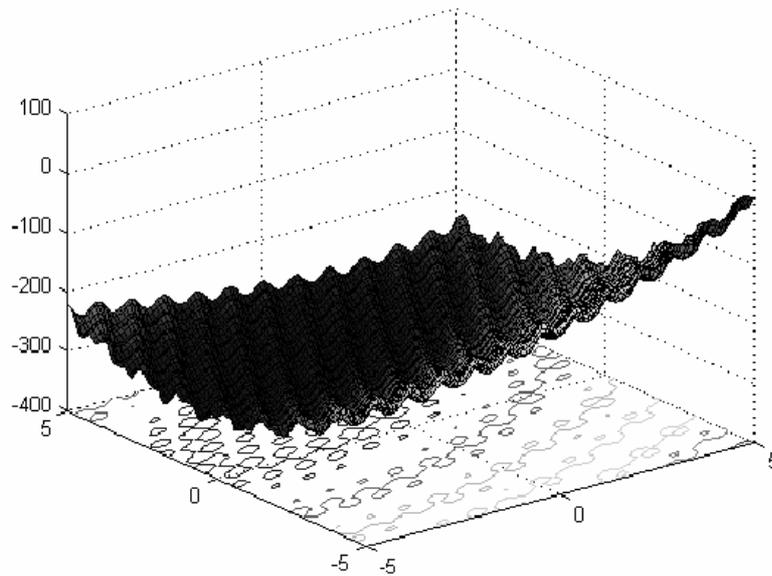
Propiedades:

- Multi-modal.
- Desplazada.
- Separable.
- Escalable.
- Un enorme número de óptimos locales.
- $x \in [-5, 5]^D$ , óptimo global  $x^* = o$ ,  $f(x^*) = f\_bias(9) = -330$ .

**F10: Función de Rastrigin Desplazada y Rotada**

$$f(x) = \sum_{i=1}^D (z_i^2 - 10\cos(2\pi z_i) + 10) + f\_bias, z = (x - o) * M, x = [x_1, x_2, \dots, x_D]$$

Donde  $M$  es una matriz de transformación lineal, con 2 de número de condición.



Propiedades:

- Multi-modal.
- Desplazada.
- Rotada.
- Separable.
- Escalable.

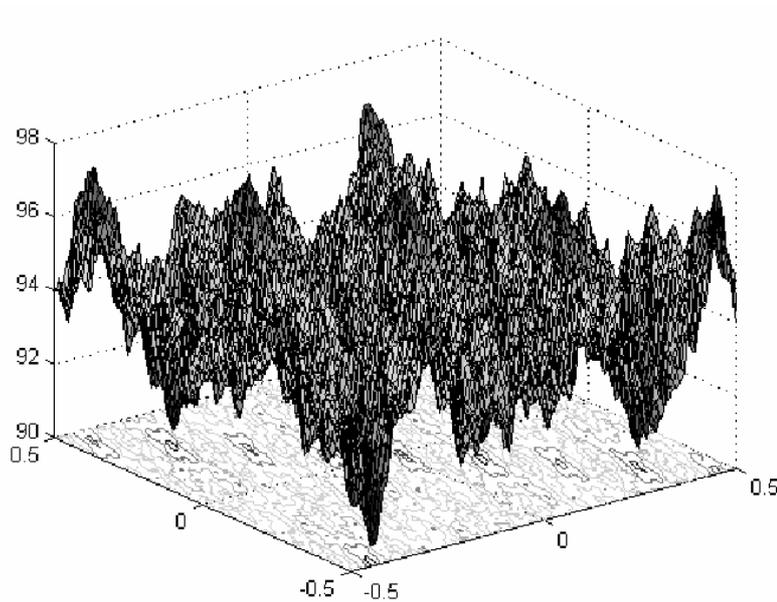
- Un enorme número de óptimos locales.
- $x \in [-5, 5]^D$ , óptimo global  $x^* = o$ ,  $f(x^*) = f\_bias(10) = -330$ .

### F11: Función de Weierstrass Desplazada y Rotada

$$f(x) = \sum_{i=1}^D \left( \sum_{k=0}^k \max[a^k \cos(2\pi b^k (z_i + 0,5))] \right) - D \sum_{k=0}^k \max[a^k \cos(2\pi b^k \cdot 0,5)] + f\_bias$$

$$a=0.5, b=3, kmax=20, z = (x - o) * M, x = [x_1, x_2, \dots, x_D]$$

Donde  $M$  es una matriz de transformación lineal, con 5 de número de condición.



Propiedades:

- Multi-modal.

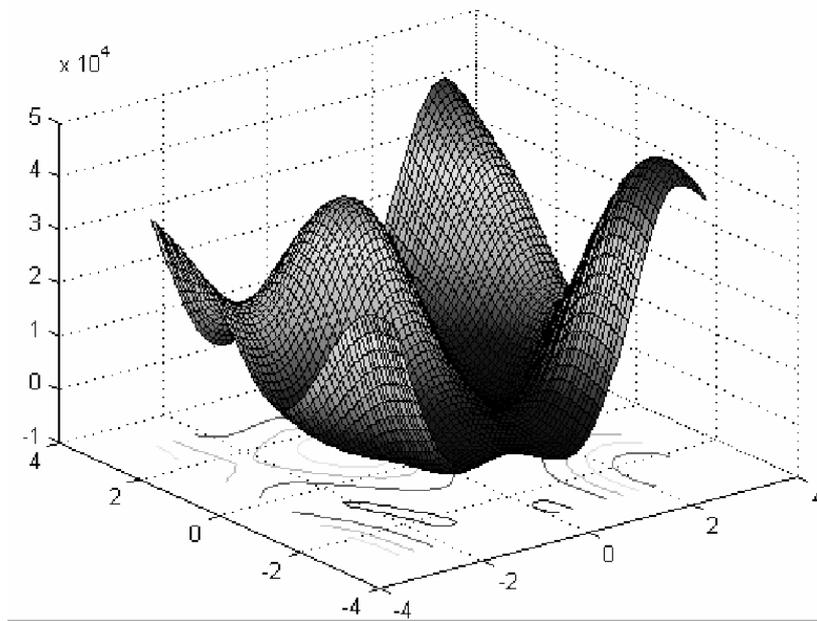
- Desplazada.
- Rotada.
- No Separable.
- Escalable.
- Continua, pero sólo diferenciable en un conjunto de puntos.
- $x \in [-0,5,0,5]^D$ , óptimo global  $x^* = o$ ,  $f(x^*) = f\_bias(11) = 90$ .

### F12: Problema 2.13 de Schwefel

$$f(x) = \sum_{i=1}^D (A_i + B_i(x))^2 + f\_bias, x = [x_1, x_2, \dots, x_D]$$

$$A_i = \sum_{j=1}^D (a_{ij} \text{sen} \alpha_j + b_{ij} \text{cos} \alpha_j), B_i = \sum_{j=1}^D (a_{ij} \text{sen} x_j + b_{ij} \text{cos} x_j)$$

Donde  $A, B$  son dos matrices  $D * D$ ,  $a_{ij}, b_{ij}$  son enteros aleatorios en el rango  $[-100, 100]$ ,  $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_D]$  son valores aleatorios en el rango  $[-\pi, \pi]$ .



Propiedades:

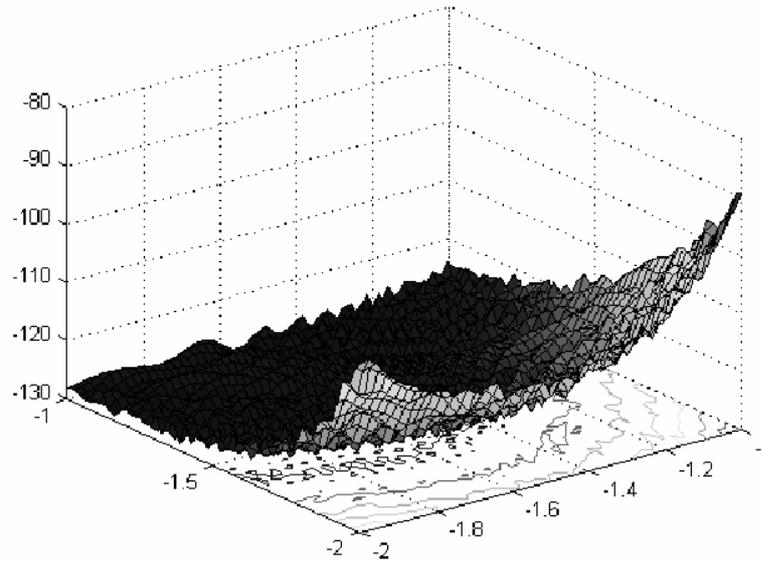
- Multi-modal.
- Desplazada.
- No Separable.
- Escalable.
- $x \in [-\pi, \pi]^D$ , óptimo global  $x^* = \alpha$ ,  $f(x^*) = f\_bias(12) = -460$ .

### F13: Funciones de Griewank y Rosenbrock Expandidas y Extendidas

F8: Función de Griewank 
$$F8(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

F2: Función de Rosenbrock 
$$F2(x) = \sum_{i=1}^D (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$$

$$f(x) = F8F2(z_1, z_2, \dots, z_D) = F8(F2(z_1, z_2)) + F8(F2(z_2, z_3)) + \dots + F8(F2(z_{D-1}, z_D)) \\ + F8(F2(z_D, z_1)) + f\_bias, z = x - o + 1, x = [x_1, x_2, \dots, x_D]$$



Propiedades:

- Multi-modal.
- Desplazada.
- No Separable.
- Escalable.
- $x \in [-5, 5]^D$ , óptimo global  $x^* = \alpha$ ,  $f(x^*) = f\_bias(13) = -130$ .

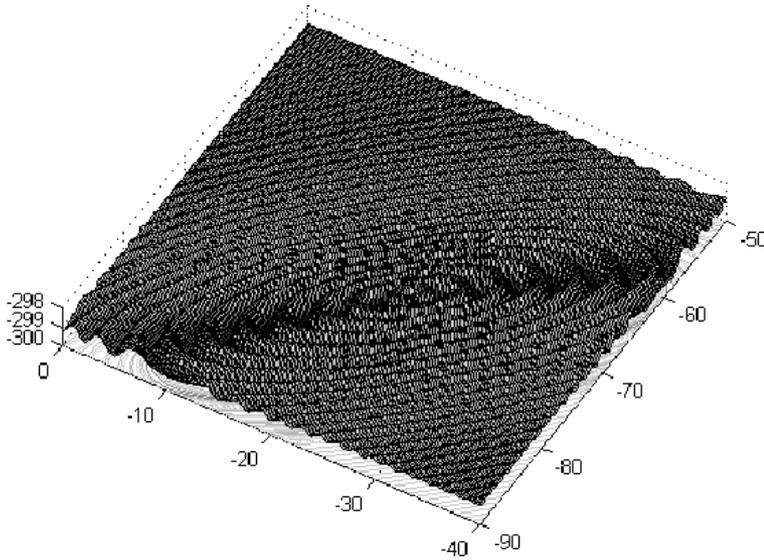
#### F14: Función Scaffer expandida, Extendida y Rotada

$$F(x, y) = 0,5 + \frac{(\text{sen}^2(\sqrt{x^2+y^2})-0,5)}{(1+0,001(x^2+y^2))^2}$$

$$f(x) = F(z_1, z_2) + F(z_2, z_3) + \cdots + F(z_{D-1}, z_D) + F(z_D, z_1) + f\_bias$$

$$z = (x - o) * M, x = [x_1, x_2, \dots, x_D]$$

Donde  $M$  es una matriz de transformación lineal, con 3 de número de condición.



Propiedades:

- Multi-modal.
- Desplazada.
- Rotada.
- No Separable.
- Escalable.
- $x \in [-100, 100]^D$ , óptimo global  $x^* = o$ ,  $f(x^*) = f\_bias(14) = -300$ .

## Cálculo de las funciones compuestas

Todas las funciones restantes, compuestas, se calculan de la siguiente forma:

$$F(x) = \sum_{i=1}^n (w_i \cdot (f'_i((x - o_i)) / \lambda_i * M_i) + bias_i) + f\_bias$$

Donde  $w_i$  son valores de peso, calculado como:

$$w_i = \exp\left(-\frac{\sum_{k=1}^D (x_k - o_{ik})^2}{2D\sigma_i^2}\right), \text{ luego normalizamos } w_i = w_i / \sum_{i=1}^n w_i.$$

Donde  $\sigma_i$  controla el rango de cobertura de  $f'_i(x)$ , un valor bajo de  $\sigma_i$  da un rango reducido de  $f_i(x)$ .

$\lambda_i$  es empleado para estirar o desplegar la función,  $\lambda_i > 1$  implica expandirla, mientras que  $\lambda_i < 1$  implica una contracción.

$o_i$  y  $bias_i$  definen la posición del óptimo global.

### F15: Función Híbrida Compuesta 1

$f_{1-2}(x)$  : Función de Rastrigin.

$$f_i(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10).$$

$f_{3-4}(x)$  : Función Weierstrass.

$f_i(x) = \sum_{i=1}^D (\sum_{k=0}^k \max[a^k \cos(2\pi b^k (x_i + 0,5))]) - D \sum_{k=0}^k \max[a^k \cos(2\pi b^k \cdot 0,5)]$ . a=0.5, b=3, kmax=20

$f_{5-6}(x)$  : Función Griewank.

$$f_i(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1.$$

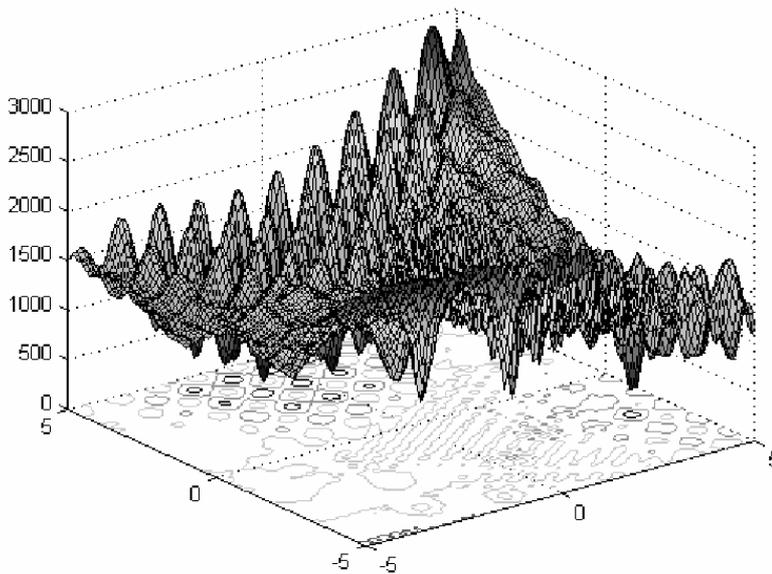
$f_{7-8}(x)$  : Función Ackley.

$$f_i(x) = -20 \cdot \exp\left(-0,2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right) + 20 + e.$$

$f_{9-10}(x)$  : Función Esfera.

$$f_i(x) = \sum_{i=1}^D x_i^2.$$

Donde  $\sigma_i = 1 \forall i = 1, 2, \dots, D$ .  $\lambda = [1, 1, 10, 10, 5/60, 5/60, 5/32, 5/32, 5/100, 5/100]$   $M_i$  son todas la matriz identidad.

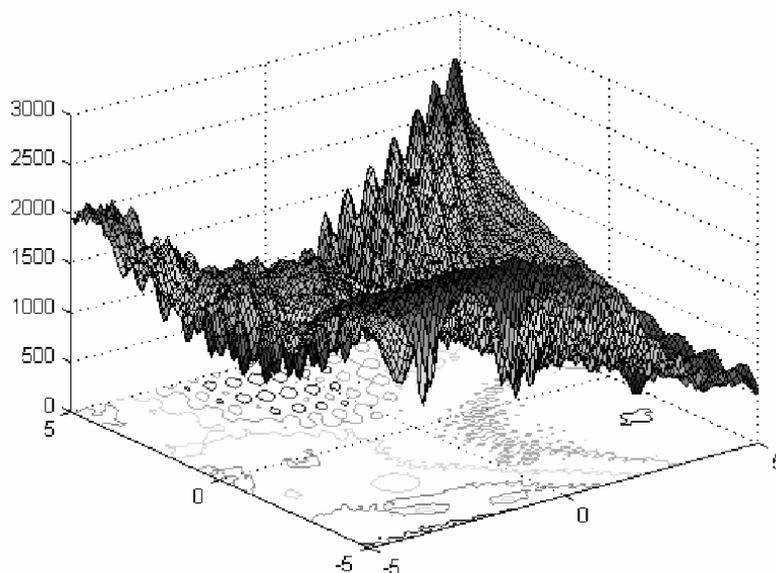


Propiedades:

- Multi-modal.
- Separable cerca del óptimo global.
- Escalable.
- Un amplio número de óptimos locales.
- Combina las propiedades de distintas funciones.
- Las funciones esféricas originan dos zonas suaves.
- $x \in [-5, 5]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) \approx f\_bias(15) = 120$ .

### F16: Función Híbrida Compuesta 1 Rotada

Toda la configuración es equivalente a la Función Híbrida Compuesta 1, a excepción de  $M_i$  que son matrices de transformación lineal.



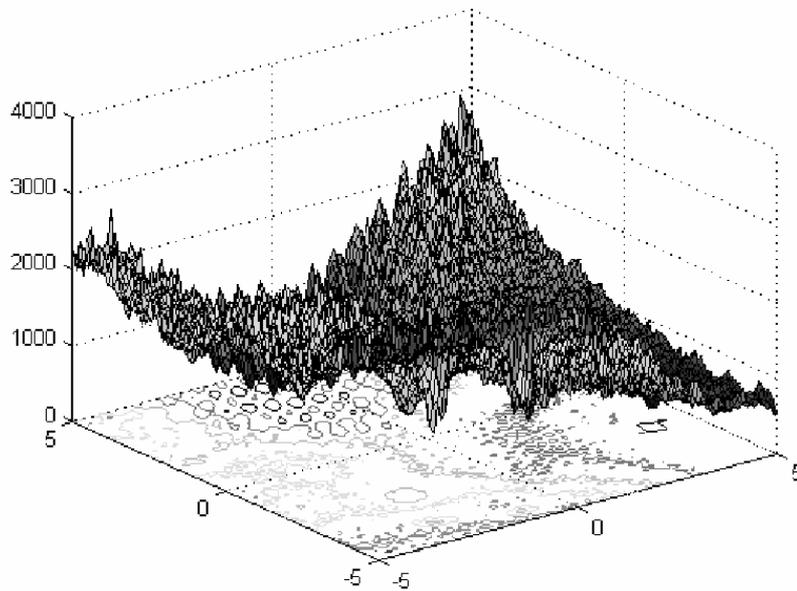
Propiedades:

- Multi-modal.
- Rotada.
- No Separable.
- Escalable.
- Un amplio número de óptimos locales.
- Combina las propiedades de distintas funciones.
- Las funciones esféricas originan dos zonas suaves.
- $x \in [-5, 5]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) \approx f\_bias(16) = 120$ .

### F17: Función Híbrida Compuesta 1 con Ruido

Sea  $G(x)$  la Función Híbrida 1 Rotada antes de añadir  $f\_bias$ , se define

$$f(x) = G(x) * (1 + 0,2|N(0, 1)|) + f\_bias$$



Propiedades:

- Multi-modal.
- Rotada.
- No Separable.
- Escalable.
- Un amplio número de óptimos locales.
- Combina las propiedades de distintas funciones.
- Las funciones esféricas originan dos zonas suaves.

■ Ruido Gaussiano.

■  $x \in [-5, 5]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) \approx f_{bias}(17) = 120$ .

## F18: Función Híbrida Compuesta 2

$f_{1-2}(x)$  : Función Ackley.

$$f_i(x) = -20 \cdot \exp\left(-0,2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right) + 20 + e.$$

$f_{3-4}(x)$  : Función de Rastrigin.

$$f_i(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10).$$

$f_{5-6}(x)$  : Función Esfera.

$$f_i(x) = \sum_{i=1}^D x_i^2.$$

$f_{7-8}(x)$  : Función Weierstrass.

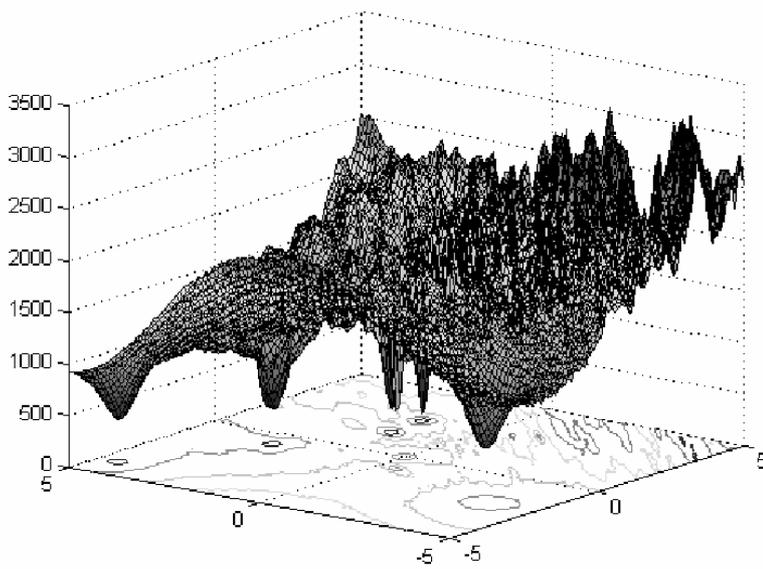
$$f_i(x) = \sum_{i=1}^D (\sum_{k=0}^k \max[a^k \cos(2\pi b^k (x_i + 0,5))]) - D \sum_{k=0}^k \max[a^k \cos(2\pi b^k \cdot 0,5)]. \quad a=0.5, b=3, kmax=20$$

$f_{9-10}(x)$  : Función Griewank.

$$f_i(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1.$$

Donde  $\sigma = [1, 2, 1.5, 1.5, 1, 1, 1.5, 1.5, 2, 2]$ .  $\lambda = [2^*5/32, 5/32, 2^*1, 1, 2^*5/100, 5/100, 2^*10, 10, 2^*5/60, 5/60]$   $M_i$  son matrices de rotación, de número de condición [2 3 2 3 2 3 20 30 200 300].

$$o = [0, 0, \dots, 0]$$



Propiedades:

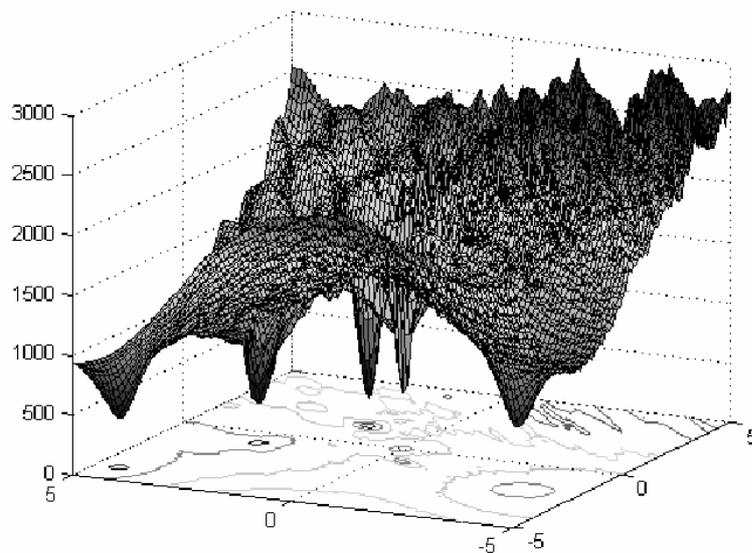
- Multi-modal.
- Rotada.
- No Separable.
- Escalable.
- Un amplio número de óptimos locales.
- Combina las propiedades de distintas funciones.
- Las funciones esféricas originan dos zonas suaves.
- Un óptimo local en el origen.
- $x \in [-5, 5]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) \approx f\_bias(18) = 10$ .

### F19: Función Híbrida Compuesta 2 con una Estrecha Base sobre el Optimo

Toda la configuración es equivalente a la Función Híbrida Compuesta 2, a excepción de:

$$\lambda = [0.1, 2, 1.5, 1.5, 1, 1, 1.5, 1.5, 2, 2]$$

$$\lambda = [0.1 * 5 / 32, 5 / 32, 2 * 1, 1, 2 * 5 / 100, 5 / 100, 2 * 10, 10, 2 * 5 / 60, 5 / 60]$$



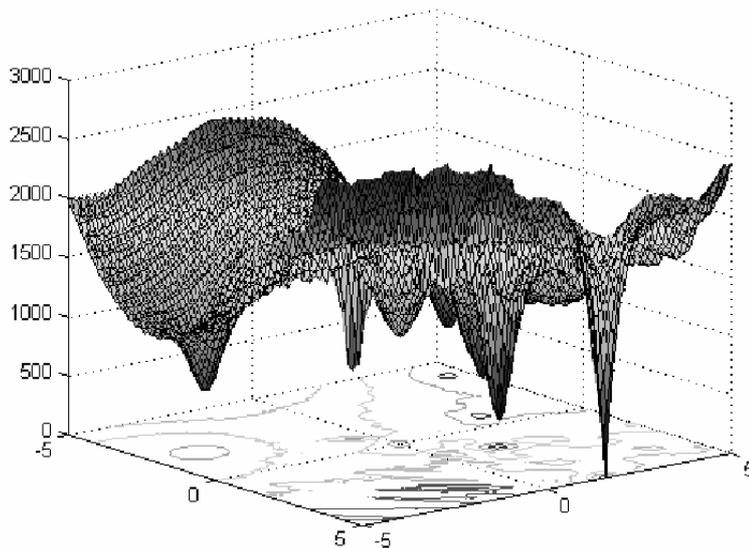
Propiedades:

- Multi-modal.
- Rotada.
- No Separable.
- Escalable.
- Un amplio número de óptimos locales.
- Combina las propiedades de distintas funciones.

- Las funciones esféricas originan dos zonas suaves.
- Un óptimo local en el origen.
- Una base de atracción estrecha cerca del óptimo global.
- $x \in [-5, 5]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) \approx f\_bias(19) = 10$ .

## F20: Función Híbrida Compuesta 2 con Optimos Globales en la Frontera

Toda la configuración es equivalente a la Función Híbrida Compuesta 2, a excepción de  $\sigma_{12j}=5$ ,  $j = 1, 2, \dots, \lfloor D/2 \rfloor$ .



Propiedades:

- Multi-modal.
- No Separable.

- Escalable.
- Un amplio número de óptimos locales.
- Combina las propiedades de distintas funciones.
- Las funciones esféricas originan dos zonas suaves.
- Un óptimo local en el origen.
- Un óptimo global en la frontera.
- Si el procedimiento de inicialización inicializa a la población en la frontera, la función será resuelta fácilmente.
- $x \in [-5, 5]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) \approx f_{bias}(20) = 10$ .

### F21: Función Híbrida Compuesta 3

$f_{1-2}(x)$ : Función Scaffer Rotada y Expandida (F6)

$$F(x, y) = 0,5 + \frac{(\text{sen}^2(\sqrt{x^2+y^2})-0,5)}{(1+0,001(x^2+y^2))^2}$$

$$f(x) = F(x_1, x_2) + F(x_2, x_3) + \cdots + F(x_{D-1}, x_D) + F(x_D, x_1)$$

$f_{3-4}(x)$  : Función de Rastrigin.

$$f_i(x) = \sum_{i=1}^D (x_i^2 - 10\cos(2\pi x_i) + 10).$$

$f_{5-6}(x)$  : Función F8F2.

$$F8(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$F2(x) = \sum_{i=1}^D (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$$

$$f_i(x) = F8(F2(x_1, x_2)) + F8(F2(x_2, x_3)) + \cdots + F8(F2(x_{D-1}, x_D)) + F8(F2(x_D, x_1))$$

$f_{7-8}(x)$  : Función Weierstrass.

$$f_i(x) = \sum_{i=1}^D (\sum_{k=0}^k \max[a^k \cos(2\pi b^k (x_i + 0,5))]) - D \sum_{k=0}^k \max[a^k \cos(2\pi b^k \cdot 0,5)]. \quad a=0.5, b=3, k_{\max}=20$$

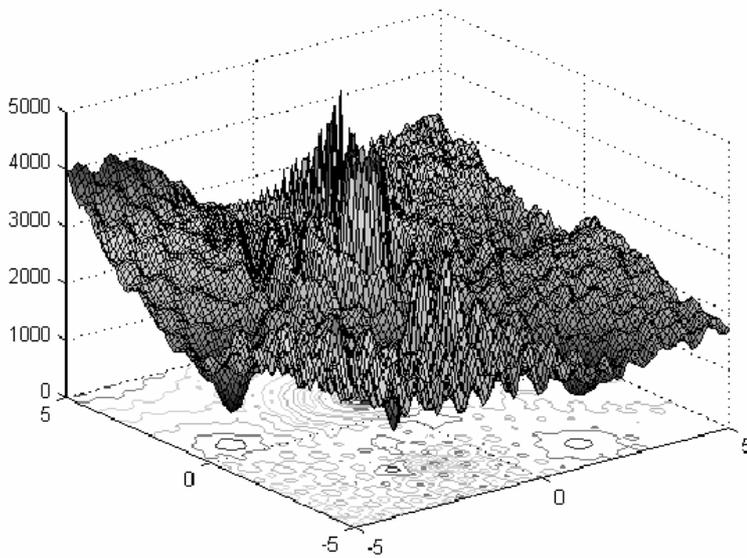
$f_{9-10}(x)$  : Función Griewank.

$$f_i(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1.$$

$$\sigma = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2]$$

$$\lambda = [5 \cdot 5/100, 5/100, 5 \cdot 1, 1, 5 \cdot 1, 1, 5 \cdot 10, 10, 5 \cdot 5/200, 5/200]$$

$M_i$  son todas matrices ortogonales.



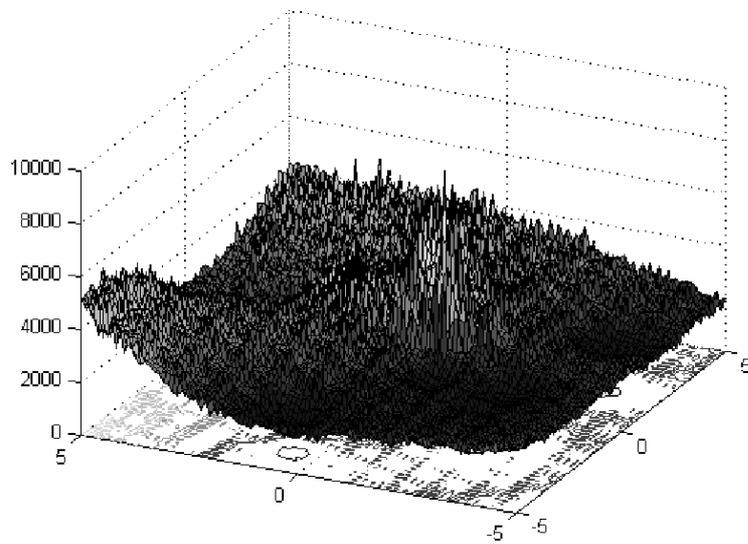
Propiedades:

- Multi-modal.
- Rotada.
- No Separable.
- Escalable.

- Un amplio número de óptimos locales.
- Combina las propiedades de distintas funciones.
- $x \in [-5, 5]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) \approx f\_bias(21) = 360$ .

### F22: Función Híbrida Compuesta 3 Altamente Condicionada

Toda la configuración es equivalente a la Función Híbrida Compuesta 3, a excepción de  $M'_i$ , cuyos valores de condicionamiento son [10, 20, 50, 100, 200, 1000, 2000, 3000, 4000, 5000].



Propiedades:

- Multi-modal.
- No Separable.
- Escalable.
- Un amplio número de óptimos locales.

- Combina las propiedades de distintas funciones.
- Un óptimo global en la frontera.
- $x \in [-5, 5]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) \approx f_{bias}(22) = 360$ .

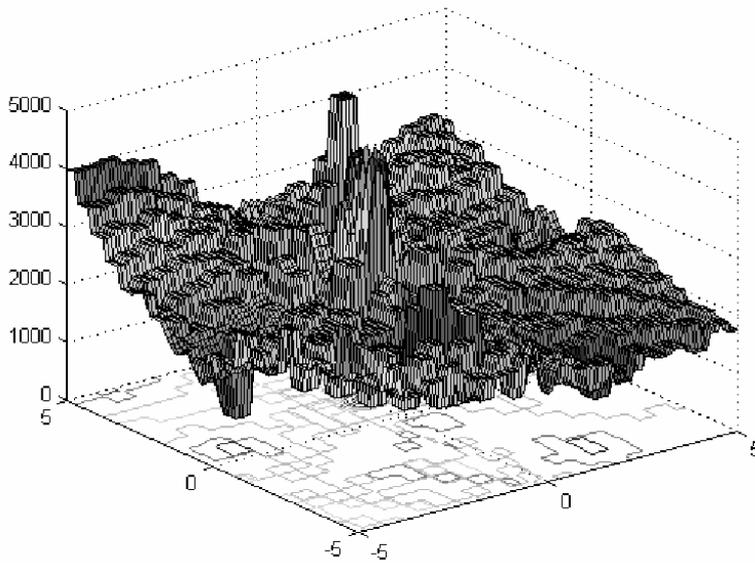
### F23: Función Híbrida Compuesta 3 Rotada y No Continúa

Toda la configuración es equivalente a la Función Híbrida Compuesta 3, a excepción de

$$x_j = \begin{cases} x_j & |x_j - o_{ij}| < 1/2 \\ \text{round}(2x_j)/2 & |x_j - o_{1j}| \leq 1/2 \end{cases}$$

$$\text{round}(x) = \begin{cases} a - 1 & \text{si } x \leq 0 \text{ y } b \geq 0,5 \\ a & \text{si } b < 0,5 \\ a + 1 & \text{si } x > 0 \text{ y } b \geq 0,5 \end{cases}$$

Donde  $a$  es la parte entera y  $b$  es la parte decimal



Propiedades:

- Multi-modal.
- No Separable.
- Escalable.
- Un amplio número de óptimos locales.
- Combina las propiedades de distintas funciones.
- No continúa.
- Un óptimo global en la frontera.
- $x \in [-5, 5]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) \approx f\_bias(23) = 360$ .

#### F24: Función Híbrida Compuesta 4

$f_1(x)$  : Función Weierstrass.

$$f_i(x) = \sum_{i=1}^D (\sum_{k=0}^k \max[a^k \cos(2\pi b^k(x_i + 0,5))]) - D \sum_{k=0}^k \max[a^k \cos(2\pi b^k \cdot 0,5)]. \quad a=0.5, b=3, kmax=20$$

$f_2(x)$ : Función Scaffer Rotada y Expandida (F6)

$$F(x, y) = 0,5 + \frac{(\text{sen}^2(\sqrt{x^2+y^2})-0,5)}{(1+0,001(x^2+y^2))^2}$$

$$f(x) = F(x_1, x_2) + F(x_2, x_3) + \dots + F(x_{D-1}, x_D) + F(x_D, x_1)$$

$f_3(x)$  : Función F8F2.

$$F8(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$F2(x) = \sum_{i=1}^D (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$$

$$f_i(x) = F8(F2(x_1, x_2)) + F8(F2(x_2, x_3)) + \dots + F8(F2(x_{D-1}, x_D)) + F8(F2(x_D, x_1))$$

$f_4(x)$  : Función Ackley.

$$f_i(x) = -20 \cdot \exp(-0,2\sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)) + 20 + e.$$

$f_5(x)$  : Función de Rastrigin.

$$f_i(x) = \sum_{i=1}^D (x_i^2 - 10\cos(2\pi x_i) + 10).$$

$f_6(x)$  : Función Griewank.

$$f_i(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1.$$

$f_7(x)$  : Función de Scaffer expandida No continúa

$$F(x, y) = 0,5 + \frac{(\text{sen}^2(\sqrt{x^2+y^2})-0,5)}{(1+0,001(x^2+y^2))^2}$$

$$f(x) = F(y_1, y_2) + F(y_2, y_3) + \dots + F(y_{D-1}, y_D) + F(y_D, y_1)$$

$$y_j = \begin{cases} x_j & |x_j| < 1/2 \\ \text{round}(2x_j)/2 & |x_j| \geq 1/2 \end{cases}$$

$f_8(x)$  : Función de Rastrigin No continúa

$$f_i(x) = \sum_{i=1}^D (y_i^2 - 10\cos(2\pi y_i) + 10).$$

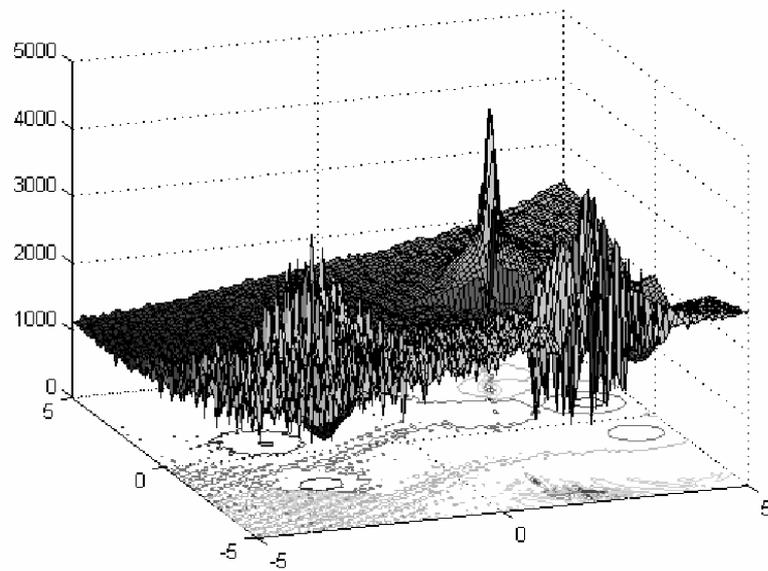
$$y_j = \begin{cases} x_j & |x_j| < 1/2 \\ \text{round}(2x_j)/2 & |x_j| \geq 1/2 \end{cases}$$

$f_9(x)$  : Función Elíptica altamente condicionada

$$f_i(x) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2$$

$f_{10}(x)$  : Función Esfera con Ruido en la Evaluación

$$f_i(x) = (\sum_{i=1}^D x_i^2)(1 + 0,1|N(0, 1)|).$$

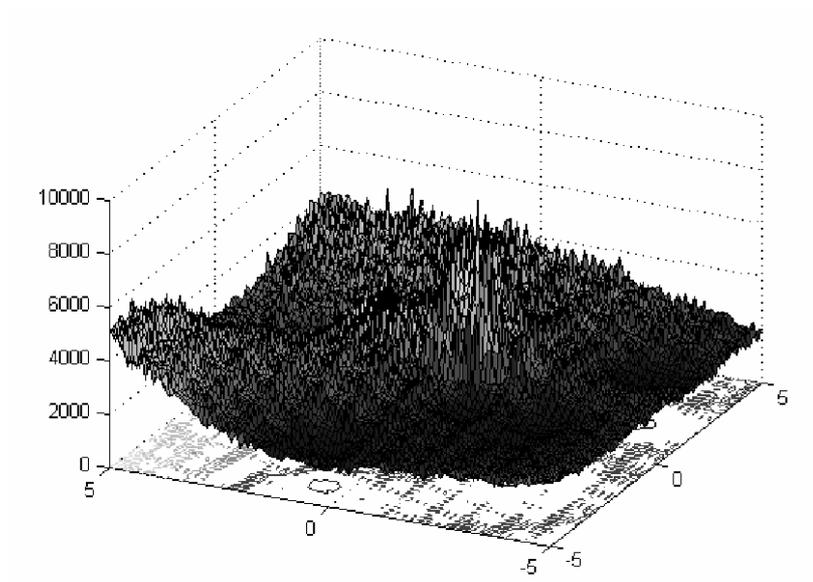


Propiedades:

- Multi-modal.
- Rotada.
- No Separable.
- Escalable.
- Un amplio número de óptimos locales.
- Combina las propiedades de distintas funciones.
- Las funciones unimodales originan zonas suaves.
- $x \in [-5, 5]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) \approx f\_bias(24) = 260$ .

### F25: Función Híbrida Compuesta 4 sin Frontera

Toda la configuración es equivalente a la Función Híbrida Compuesta 4, con la diferencia de que no hay un rango de búsqueda delimitado.



Propiedades:

- Multi-modal.
- Rotada.
- No Separable.
- Escalable.
- Un amplio número de óptimos locales.
- Combina las propiedades de distintas funciones.
- Las funciones unimodales originan zonas suaves.
- Un óptimo global en la frontera.
- Sin límites de búsqueda.
- $x \in [-5, 5]^D$ , óptimo global,  $x^* = o$ ,  $f(x^*) \approx f\_bias(25) = 260$ .



# Bibliografía

- [Ack87] Ackley D. H. (1987) A Connectionist Machine for Genetic Hillclimbing. *Kluwer Academic Publishers* .
- [AH05a] Auger A. y Hansen N. (2005) A Restart CMA Evolution Strategy with Increasing Population Size. En *2005 IEEE Congress on Evolutionary Computation*, páginas 1769–1776.
- [AH05b] Auger A. y Hansen N. (2005) Performance Evaluation of an Advanced Local Search Evolutionary Algorithm. En *2005 IEEE Congress on Evolutionary Computation*, páginas 1777–1784.
- [Bö92] Bäck T. (1992) Self-adaptation in Genetic Algorithms. *Proc of the First European Conf. on Artificial Life* páginas 263–271.
- [Bak85] Baker J. E. (1985) Adaptive Selection Methods for Genetic Algorithms. En *Proceedings of the 1st International Conference on Genetic Algorithms*, páginas 101–111. Lawrence Erlbaum Associates, Inc.
- [Bal96] Baldwin M. J. (1896) A new factor in evolution. *The American Naturalist* 30(354): 441–451.
- [BBTZ04] Bambha N. K., Bhattacharyya S. S., Teich J., y Zitzler E. (2004) Systematic Integration of Parameterized Local Search into Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation* 8(2): 137–155.
- [BD01] Beyer H. y Deb K. (2001) On Self-adaptive Features in Real-parameter Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation* 3(5): 250–270.

- [Bey01] Beyer H.-G. (2001) *The Theory of Evolution Strategies*. Springer.
- [BFM97] Bäck T., Fogel D. B., y Michalewicz Z. (Eds.) (1997) *Handbook of Evolutionary Computation*. IOP Publishing Ltd., Bristol, UK.
- [BRS66] Bremermann H., Rogson M., y Salaff S. (1966) *Natural Automata and Useful Simulations*, chapter Global properties of evolution processes, páginas 3–41. Washington DC: Spartan.
- [BSCG05] Ballester P. J., Stephenson J., Carter J., y Gallager K. (2005) Real-Parameter Optimization Performance Study on the CEC-2005 benchmark with SPC-PNX. En *2005 IEEE Congress on Evolutionary Computation*, páginas 498–505.
- [CE05] Craenen B. y Eiben A. (2005) Hybrid Evolutionary Algorithms for Constraint Satisfaction Problems: Memetic Overkill? En *The IEEE Congress on Evolutionary Computation*, volumen 4, páginas 1922–1928.
- [Col68] Colville A. (1968) A Comparative Study of Nonlinear Programming Code. *Technical Report N 320-2949* .
- [CS03] Chelouha R. y Siarry P. (2003) Genetic and Nelder-Mead Algorithms Hybridized for a more Accurate Global Optimization of Continuous Multim minima Functions. *European Journal of Operational Research* 148(2): 335–348.
- [CSM04] Cioppa A. D., Stefano C. D., y Marcelli A. (2004) On the Role of Population Size and Niche Radius in Fitness Sharing. *IEEE Transactions on Evolutionary Computation* 8(5).
- [CV99] Cedeno W. y Vemuri V. R. (1999) Analysis of Speciation and Nicheing in the Multi-niche Crowding GA. *heoretical computer science* 229(1): 177–197.
- [DAJ02] Deb K., Anand A., y Joshi D. (2002) A computationally efficient evolutionary algorithm for real-parameter evolution. *Evolutionary Computation Journal* 10(4): 371–395.
- [Dar59] Darwin C. (1859) *On the Origin of Species*. London: John Murray.

- [Dar76] Darwin R. (1976) *The Selfish Gene (El Gen Egoísta)*. Oxford University Press, Oxford.
- [Deb01] Deb K. (2001) *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons.
- [Dem06] Demšar J. (2006) Statistical Comparisons of Classifiers Over Multiple Data Sets. *Journal of Machine Learning Research* 7: 1–30.
- [dJ75] de Jong K. (1975) *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan.
- [DJ02] Deb K. y Joshi D. (2002) A computationally efficient evolutionary algorithm for real-parameter evolution. *Evolutionary Computation Journal* 10(4): 371–395.
- [DNVHW03] Dietzfelbinger M., Naudts B., Van-Hoyweghen C., y Wegener I. (2003) The Analysis of a Recombinative Hill-Climber on H-IFF. *IEEE Transactions on Evolutionary Computation* 7(5): 417–423.
- [DS04] Dorigo M. y Stützle T. (2004) *Ant Colony Optimization*. MIT Press.
- [dSCCK02] dos Santos Coelho L., Coelho A. A. R., y Krohling R. A. (2002) Parameters Tuning of Multivariable Controllers Based on Memetic Algorithms: Fundamentals and Application. En *2002 IEE International Symposium on Intelligente System*, páginas 27–30.
- [ECS93] Eshelman L., Caruana A., y Schaffer J. (1993) Real-Coded Genetic Algorithms and Interval-Schemata. *Foundation of Genetic Algorithms* 2: 187–202.
- [EHM99] Eiben A., Hinterding R., y Michalewicz Z. (1999) Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation* 3(2).
- [EMS97] Eshelman L. J., Mathias K. E., y Schaffer J. D. (1997) Convergence controlled variation. *Foundations of Genetic Algorithms* páginas 203–224.
- [ES91] Eshelman L. y Schaffer J. (1991) Preventing Premature Convergence in Genetic Algorithms by Preventing Incest. *Proc. of the Fourth Int. Conf. on Genetic Algorithm* páginas 265–283.

- [ES93] Eshelman L. J. y Schaffer J. D. (1993) Real-coded Genetic Algorithms in Genetic Algorithms by Preventing Incest. *Foundation of Genetic Algorithms 2* páginas 187–202.
- [ES98] Eiben A. y Schippers C. (1998) On Evolutionary Exploration and Exploitation. *Fund. Inf.* 35: 1–16.
- [ES03] Eiben A. y Smith J. (2003) *Introduction to Evolutionary Computing*. Springer.
- [Esh91] Eshelman L. (1991) The CHC Adaptive Search Algorithm. How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. in *Foundations of Genetic Algorithms* páginas 265–283.
- [FR01] Fernandes C. y Rosa A. (2001) A Study of non-Random Matching and Varying Population Size in Genetic Algorithm using a Royal Road Function. *Proc. of the 2001 Congress on Evolutionary Computation* páginas 60–66.
- [Fri40] Friedman M. (1940) A Comparison of Alternative Tests of Significance for the Problem of m Ranking. *Annals of Mathematical Statistics* 11: 86–92.
- [GD91] Goldberg D. E. y Deb K. (1991) A Comparative Analysis of Selection Schemes used in Genetic Algorithms. *Foundations of Genetic Algorithms* páginas 69–93.
- [GK03] Glover F. y Kochenberger G. (Eds.) (2003) *Handbook of Metaheuristics*. Kluwer Academic Publishers.
- [GL97] Glover F. y Laguna M. (1997) *Tabu Search*. Kluwer Academic Publishers.
- [Glo77] Glover F. (1977) Heuristics for Integer Programming Using Surrogate Constraints. *Decision Sciences* 8(1), 156–166. *Decision Sciences* 8(1): 156–166.
- [Glo98a] Glover F. (1998) A Template for Scatter Search and Path Relinking. *Artificial Evolution, Lecture Notes in Computer Science* Springer: 13–54.

- [Glo98b] Glover F. (1998) Genetic Algorithms, Evolutionary Algorithms and Scatter Search: Changing Tides and Untapped Potentials. *INFORMS Computer Science Technical Section Newsletter* 19(1): 7–14.
- [GML05] Garcia-Martinez C. y Lozano M. (2005) Hybrid Real-Coded Genetic Algorithms with Female and Male Differentiation. En *2005 IEEE Congress on Evolutionary Computation*, páginas 896–903.
- [GMLH07] Garcia S., Molina D., Lozano M., y Herrera F. (2007) An experimental study about the use of non-parametric tests for analysing the behaviour of evolutionary algorithms in optimization problems. *In Spanish*. En *MAEB 2007 (Quinto congreso español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados)*.
- [Gol89] Goldberg D. (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. *Addison-Wesley, Reading, MA*.
- [Gos08] Gosset W. (1908) The probable error of a mean. *Biometrika* 6(1): 1–25.
- [Gri81] Griewangk A. O. (1981) Generalized descent of global optimization. *Journal of Optimization Theory and Applications* 34: 11–39.
- [GW97] Goldberg D. y Wang L. (1997) Adaptive niching via coevolutionary sharing. *Genetic Algorithms in Engineering and Computer Science* páginas 21–38.
- [Han05a] Hansen N. (2005) Compilation of Results on the CEC Benchmark Function Set. Technical report, Institute of Computational Science, ETH Zurich, Switerland. available as [http://www.ntu.edu.sg/home/epsugan/index\\_files/CEC-05/compareresults.pdf](http://www.ntu.edu.sg/home/epsugan/index_files/CEC-05/compareresults.pdf).
- [Han05b] Hansen N. (2005) Compilation of Results on the CEC Benchmark Function Set. En *2005 IEEE Congress on Evolutionary Computation*.
- [Har94] Hart W. (1994) *Adaptive Global Optimization With Local Search*. PhD thesis, Univ. California, San Diego, CA.

- [Har95] Harik G. (1995) Finding Multimodal Solutions Using Restricted Tournament Selection. *Proc. 6th Int. Conf. Genetic Algorithms* páginas 24–31.
- [HK01] Hansen N. y Kern S. (2001) Evaluating the CMA Evolution Strategy on Multimodal Test Functions. *Evolutionary Computation* 9(2): 159–195.
- [HK04] Hansen N. y Kern S. (2004) Evaluating the CMA Evolution Strategy on Multimodal Test Functions. En *Eighth International Conference on Parallel Problem Solving from Nature PPSN VIII, Proceedings*, páginas 282–291. Springer, Berlin.
- [HKS04] Hart W., Krasnogor N., y Smith J. (2004) *Recent Advances in Memetic Algorithms*. Springer, Berlin, Heidelberg, New York.
- [HL00] Herrera F. y Lozano M. (2000) Gradual Distributed Real-coded Genetic Algorithms. *IEEE Transactions on Evolutionary Computation* 4(1): 43–63.
- [HLM06] Herrera F., Lozano M., y Molina D. (2006) Continuous Scatter Search: An Analysis of the Integration of Some Combination Methods and Improvement Strategies. *European Journal of Operational Research* 169(2): 450–476.
- [HLS03] Herrera F., Lozano M., y Sanchez A. M. (2003) A Taxonomy for the Crossover Operator for Real-coded Genetic Algorithms. *International Journal of Intelligent Systems* 18(3): 204–217.
- [HLV98] Herrera F., Lozano M., y Verdegay J. L. (1998) Tackling Real-coded Genetic Algorithms: Operators and Tools for the Behavioral Analysis. *Artificial Intelligence Reviews* 12(4): 265–319.
- [HM01] Hansen P. y Mladenović N. (2001) Variable Neighborhood Search: Principles and Applications. *European Journal of Operational Research* 130: 449–467.
- [HMK03] Hansen N., Müller S., y Koumoutsakos P. (2003) Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation* 1(11): 1–18.

- [HO96] Hansen N. y Ostermeier A. (1996) Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation. En *Proceeding of the IEEE International Conference on Evolutionary Computation (ICEC '96)*, páginas 312–317.
- [HO01] Hansen N. y Ostermeier A. (2001) Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation* 9(2): 159–195.
- [Hol79] Holm S. (1979) A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6: 65–70.
- [HRBM00] Hart W. E., Rosin C., Belew R. K., y Morris G. M. (2000) Improved evolutionary hybrids for flexible ligand docking in autodocks. En *Proc Intl Conference on Optimization in Computational Chem and Molec Bio.*, páginas 209–230.
- [IHR07] Igel C., Hansen N., y Roth S. (2007) Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation* 15(1): 1–28.
- [IM03] Ishibuchi H. y Murata T. (2003) Balance Between Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling. *IEEE Transactions on Evolutionary Computation* 7(2): 204–223.
- [IN04] Ishibuchi H. y Narukawa K. (2004) Some Issues on the Implementation of Local Search in Evolutionary Multiobjective Optimization. En *Genetic Evolutionary Computation Conference (GECCO) 2004*, páginas 1246–1258.
- [IY03] Ishibuchi H. y Yoshida T. (2003) Balance Between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling. *IEEE Transactions on Evolutionary Computation* 7(2).
- [IYM02] Ishibuchi H., Yoshida T., y Murata T. (2002) Balance Between Genetic Search and Local Search in Hybrid Evolutionary Multi-Criterion Optimization Algorithms. En *Genetic and Evolutionary Computation Conference (GECCO) 2002*, páginas 1301–1308.

- [Jon95] Jones T. (1995) Crossover, macromutation, and population-based search. En *Proc. of the Sixth International Conference on Genetic Algorithms*, páginas 73–80. Morgan Kaufmann Publishers, San Mateo, California.
- [JS92] Jong K. A. D. y Spears W. M. (1992) A formal analysis of the role of multi-point crossover in genetic algorithms. *Annals of Mathematics and Artificial Intelligence* 5(1): 1–26.
- [KE95] Kennedy J. y Eberhart R. (1995) Particle Swarm Optimization. En *IEE International Conference on Neural Networks*, páginas 1942–1948.
- [Kem96] Kemenad C. H. M. (1996) Cluster evolution strategies, enhancing the sampling density function using representatives. En *Proc. of the Third IEEE Conference on Evolutionary Computation*, páginas 637–642. IEEE Press, Piscataway, New Jersey.
- [Ken99] Kennedy J. (1999) Small worlds and mega-minds: effects of neighborhood topology on Particle Swarm Performance. En *Congress on Evolutionary Computation*, páginas 1931–1938.
- [KGV83] Kirkpatrick S., Gelatt C., y Vecchi M. (1983) Optimization by Simulated Annealing. *Science* 220: 671–680.
- [Kit01] Kita H. (2001) A Comparison Study of Self-adaptation in Evolutionary Strategies and Real-coded Genetic Algorithms. *Evolutionary Computation Journal* 9(2): 223–241.
- [Koz92] Koza J. (1992) *Genetic programming: On the programming of computers by means of natural selection*. Cambridge, MA: MIT Press.
- [KPTP01] Kazarlis S., Papadakis S., Theocharis J., y Petridis V. (2001) Microgenetic algorithms as generalized hill-climbing operator for genetic algorithms optimization. *IEEE Transactions on Evolutionary Computation* 5(3): 204–217.
- [Kra02] Krasnogor N. (2002) *Studies on the Theory and Design Space of Memetic Algorithms*. PhD thesis, Faculty Computation, Math. Eng, Univ. West of England, Bristol U. K.

- [KS00] Krasnogor N. y Smith J. E. (2000) A Memetic Algorithm with Self-Adapting a Local Search: TSP as a Case Study. *Proceedings of the 2000 International Conference on Genetic and Evolutionary Computation* páginas 987–994.
- [KS05] Krasnogor N. y Smith J. (2005) A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issue. *IEEE Transactions on Evolutionary Computation* 9(5): 474–488.
- [Lan98] Land M. S. (1998) *Evolutionary Algorithms with Local Search for Combinational Optimization*. PhD thesis, Univ. California, San Diego, CA.
- [LCC<sup>+</sup>05] Liu B.-F., Chen H.-M., Chen J.-H., Hwang S.-F., y Ho S.-Y. (2005) Meswarm: Memetic particle swarm optimization. En *Genetic Evolutionary Computation Conference (GECCO) 2005*, páginas 267–268.
- [LHKM04] Lozano M., Herrera F., Krasnogor N., y Molina D. (2004) Real-coded Memetic Algorithms with Crossover Hill-climbing. *Evolutionary Computation* 12(2): 273–302.
- [LLIB06] Lozano J., Larrañaga P., Inza I., y Bengotxea E. (2006) *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. Springer Verlag.
- [LM03] Laguna M. y Martí R. (2003) *Scatter Search. Methodology and Implementation in C*. Kluwer Academic Publishers.
- [LP02] Langdon W. B. y Poli R. (2002) *Foundations of Genetic Programming*. Springer.
- [LQSB03] Liang J., Quin A., Suganthan P., y Baskar S. (2003) Comprehensive Learning Comprehensive Particle Swarm Optimizer. *IEEE Transactions on Evolutionary Computation* 10(3): 281–295.
- [LS05] Liang J. y Suganthan P. (2005) Dynamic Multi-swarm Particle Swarm Optimizer with Local Search. En *2005 IEEE Congress on Evolutionary Computation*, páginas 522–528.

- [LW02] Lin C.-Y. y Wu W.-H. (2002) Niche Identification Techniques in Multimodal Genetic Search with Sharing Scheme. *Advances in Engineering Software* 33: 779–791.
- [Mah92] Mahfoud S. (1992) Crowding and Preselection Revised. *Parallel Problem Solving from Nature* 2 páginas 27–36.
- [MBM05] Moscato P., Berretta R., y Mendes A. (2005) A New Memetic Algorithm for Ordering Datasets: Applications in microarray analysis. En *Proceedings of the MIC 2005 - The 6th Metaheuristics International Conference*, páginas 695–700.
- [MC03] Moscato P. y Cotta C. (2003) *Handbook of Metaheuristics*, chapter A Gentle Introduction to Memetic Algorithms, páginas 105–144. Kluwer Academic Publishers, Boston MA.
- [Mer00] Merz P. (2000) *Memetic Algorithms for Combinational Optimization Problems: Fitness Landscapes and Effective Search Strategies*. PhD thesis, Gesamthochschule Siegen.
- [Mer04] Merz P. (2004) NK-Fitness Landscapes and Memetic Algorithms with Greedy Operators and k-opt Local Search. *Recent Advances in Memetic Algorithms* 166: 209–228.
- [MF99] Merz P. y Freisleben B. (1999) *Fitness Landscapes and Memetic Algorithm Design*. D. Corne, M. Dorigo and F. Glover, editors. McGraw-Hill, London.
- [MF00] Merz P. y Freisleben B. (2000) Fitness Landscape Analysis and Memetic Algorithms for the Quadratic Assignment Problem. *IEEE Transactions on Evolutionary Computation* 4(4): 337–352.
- [MHL05a] Molina D., Herrera F., y Lozano M. (2005) Adaptive Local Search Parameters for Real-Coded Memetic Algorithms. En *2005 IEEE Congress on Evolutionary Computation*, páginas 888–895.
- [MHL05b] Molina D., Herrera F., y Lozano M. (2005) Adaptive Local Search Parameters for Real-Coded Memetic Algorithms. En *2005 IEEE Congress on Evolutionary Computation*, páginas 888–895.
- [MI95] Murata T. y Ishibuchi H. (1995) Moga: multi-objective genetic algorithms. *Evolutionary Computation* 1: 289–294.

- [Mic92] Michalewicz Z. (1992) *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin.
- [Mic96] Michalewicz Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd edition, Springer-Verlag, Berlin.
- [MK04] Merz P. y Katayama K. (2004) Memetic Algorithms for the Unconstrained Binary Quadratic Programming Problem. *Bio System* 78(1-3): 99–18.
- [MLG06] Martí R., Laguna M., y Glover F. (2006) Principles of Scatter Search. *Special issue on scatter search and path relinking. European Journal of Operational Research* 169: 359–372.
- [Mos89] Moscato P. (1989) On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Toward Memetic Algorithms. Technical report, Caltech Concurrent Computation Program. California Institute of Technology, Pasadena.
- [Mos99] Moscato P. A. (1999) *Memetic Algorithms: a Short Introduction*. D. Corne, M. Dorigo and F. Glover, editors. McGraw-Hill, London.
- [MP03] Moscato P. y Porras C. C. (2003) Una introducción a los algoritmos meméticos. *Inteligencia Artificial. Revista Iberoamericana de IA* 19: 131–148.
- [MSB91] Mühlenbein H., Schomisch M., y Born J. (1991) The parallel genetic algorithm as function optimizer. En *Proc. In Fourth International Conference on Genetic Algorithms*, páginas 271–278. Morgan Kaufmann Publishers, San Mateo, California.
- [MSV93] Mühlenbein H. y Schierkamp-Voosen D. (1993) Predictive Models for the Breeding Genetic Algorithm in Continuous Parameter Optimization. *Evolutionary Computation* 1: 25–49.
- [NI05] Noman N. y Iba H. (2005) Enhancing differential evolution performance with local search for high dimensional function optimization. En *GECCO*, páginas 967–974.
- [NK97] Nagata Y. y Kobayashi S. (1997) Edge assembly crossover: a high-power genetic algorithm for the traveling salesman problem. En

- Proc. of the Seventh International Conference on Genetic Algorithms*, páginas 450–457.
- [NM65] Nelder J. y Mead R. (1965) A simplex method for functions minimizations. *Computer Journal* 7(4): 308–313.
- [NS01] Nomura T. y Shimohara K. (2001) An Analysis of Two-parent Recombinations for Real Valued Chromosomes in an Infinite Population. *Evolutionary Computation Journal* 3(9): 283–308.
- [OK04] Ong Y. S. y Keane A. J. (2004) Meta-Lamarckian Learning in Memetic Algorithms. *IEEE Transactions on Evolutionary Computation* 4(2): 99–110.
- [OLZW06] Ong Y., Lim M.-H., Zhu N., y Wong K. (2006) Classification of Adaptive Memetic Algorithms: a Comparative Study. *IEEE Transactions on System, Man, and Cybernetic* 36(1): 141–152.
- [OO95] O'Reilly U. M. y Oppacher F. (1995) Hybridized crossover-based search techniques for program discovery. En *IEEE International Conference on Evolutionary Computation*, páginas 573–578. Morgan Kaufmann Publishers, San Mateo, California.
- [PGB01] Parthasarathy P., Goldberg D. E., y Burns S. (2001) Tackling multimodal problems in hybrid genetic algorithms. En *Proc. of the Genetic and Evolutionary Computation Conference 2001*, página 775. Morgan Kaufmann Publishers, San Mateo, California.
- [Pos05] Posik P. (2005) Real Parameter Optimisation Using Mutation Step Co-Evolution. En *2005 IEEE Congress on Evolutionary Computation*, páginas 872–879.
- [PRL05] Price K. V., Rainer R. M., y Lampinen J. A. (2005) *Differential Evolution: A Practical Approach to Global Optimization*. Springer.
- [PTVF92] Press W. H., Teukolsky S. A., Vetterling W. T., y Flannery B. P. (1992) *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA.
- [QS05] Qin A. y Suganthan P. (2005) Self-adaptive Differential Evolution Algorithm for Numerical Optimization. En *2005 IEEE Congress on Evolutionary Computation*, páginas 1785–1791.

- [RC97] Reynolds R. C. y Chung C. (1997) Knowledge-based self-adaption in evolutionary programming using cultural algorithms. *Proc. of 1997 Int. Conf. on Evolutionary Computation* páginas 71–76.
- [RKP05] Ronkkonen J., Kukkonen S., y Price K. (2005) Real-Parameter Optimization with Differential Evolution. En *2005 IEEE Congress on Evolutionary Computation*, volumen 1, páginas 506–513.
- [SB00] Seront G. y Bersini H. (2000) A new ga-local search hybrid for continuous optimization based on multi level single linkaga clustering. En *Proc. of the Genetic and Evolutionary Computation Conference 2000*, páginas 90–95.
- [Sch81] Schewefel H. P. (1981) *Numerical Optimization of Computer Models*. Wiley, Chichester.
- [SE98] Shi Y. y Eberhart C. (1998) A Modified Particle Swarm Optimizer. En *Congress on Evolutionary Computation*, páginas 69–73.
- [SEAL04] Shahidi N., Esmailzadeh H., Abdollahi M., y Lucas E. E. C. (Dec. 2004) Self-adaptive Memetic Algorithm: an Adaptive Conjugate Gradient Approach. En *IEEE Conference on Cybernetics and Intelligent Systems*, volumen 1, páginas 6–11.
- [SHL<sup>+</sup>05] Suganthan P., Hansen N., Liang J. J., Deb K., Chen Y.-P., Auger A., y Tiwari S. (May 2005) Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Technical report, Nanyang Technological University. Also available as <http://www.ntu.edu.sg/home/EPNSugan/>.
- [Sin99] Sinha A. (1999) *Designing Efficient Genetic and Evolutionary Algorithm Hybrid*. PhD thesis, Indian Institute of Technology, Kharagpur.
- [Smi03] Smith J. (2003) Co-evolving Memetic Algorithms: a Learning Approach to Robust Scalable Optimisation. En *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*, volumen 1, páginas 498–505.
- [SP95] Storn R. y Price K. (1995) Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical Report TR-95-012* .

- [SP97] Storn R. y Price K. (1997) Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization* 11: 341–359.
- [Spe95] Spears W. M. (1995) Adapting crossover in evolutionary algorithms. En *Proc. of the Evolutionary Programming Conference 1995*, páginas 367–384.
- [SS06] Söensen K. y Sevaux M. (2006) MA—PM: Memetic Algorithm with Population Management. *Computers & Operational Research* 33(5): 1214–1225.
- [STCSQ07] Sch utze O., Talbi E.-G., Coello C. C., y Santana-Quintero L. V. (2007) A Memetic PSO Algorithm for Scalar Optimization Problems. En *Proceeding of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, páginas 128–134.
- [STD05] Sinha A., Tiwari S., y Deb K. (2005) A Population-Based, Steady-State Procedure for Real-Parameter Optimization. En *2005 IEEE Congress on Evolutionary Computation*, volumen 1, páginas 514–521.
- [SW81] Solis F. J. y Wets R. J. (1981) Minimization by Random Search Techniques. *Mathematical Operations Research* 6: 19–30.
- [SYK96] Satoh H., Yamamura M., y Kobayashi S. (1996) Minimal generation gap model for GAs considering both exploration and exploitation. En *Proceeding of the 4th International Conference on Soft Computing*, páginas 494–497.
- [Sys89] Syswerda G. (1989) Uniform Crossover in Genetic Algorithms. En Schaffer J. D. (Ed.) *Proc. of the Thrid Int. Conf. on Genetic Algorithms*, páginas 2–9. Morgan Kaufmann Publishers, San Mateo.
- [TA89] Törn A. y Antanas . Z. (1989) Global Optimization. *Lecture Note in Computer Science* 350.
- [Tal02] Talbi E. (2002) A Taxonomy of Hybrid Metaheuristics. *Journal of Heuristics* 8: 541–564.

- [TF93] Tsutsui S. y Fujimoto Y. (1993) Forking genetic algorithm with blocking and shrinking modes. *Proc. 5th Int. Conf. on Genetic Algorithms* páginas 206–213.
- [TK02] Trafalis T. y Kasap S. (2002) A Novel Metaheuristics Approach for Continuous Global Optimisation. *Journal of Global Optimization* 23: 171–190.
- [TLOE06] Tang J., Lim M., Ong Y., y Er M. (2006) Parallel memetic algorithm with selective local search for large scale quadratic assignment problems. *International Journal of Innovative Computing, Information and Control* 2(6).
- [Tur96] Turney P. (1996) Myths and Legends of the Baldwin Effect. En Fogarty T. y Venturini G. (Eds.) *Proceedings of the ICML-96 (13th International Conference on Machine Learning)*, páginas 135–142.
- [WBG95] Whitley D., Bereridge R., Graves C., y Mathias K. (1995) Test driving three 1995 genetic algorithms: New test functions and geometric matching. *J. Heuristic* 1: 77–104.
- [WG03] Whitley D. y Garrett D. (2003) Quad search and hybrid genetic algorithm. En *Foundation of Genetic Algorithm (FOGA): Eight International Workshop*, páginas 1469–1480.
- [Whi89] Whitley D. (1989) The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best. *Proc. of the Third Int. Conf. on Genetic Algorithms* páginas 116–121.
- [WM97] Wolpert D. H. y Macready W. G. (1997) No Free Lunch Theorems for Optimization. *IEEE Transactions of Evolutionary Computation* 1: 67–82.
- [YG05] Yuan B. y Gallagher M. (2005) Experimental Results for the Special Session on Real-Parameter Optimization at CEC 2005: A Simple, Continuous EDA. En *2005 IEEE Congress on Evolutionary Computation*, páginas 1792–1799.
- [YI01] Yagiura M. y Ibaraki T. (2001) On Metaheuristic Algorithms for Combinational Operator Problems. *Systems Computation* 32(3): 33–55.

- 
- [YK00] Yang J. M. y Kao C. Y. (2000) Integrating adaptive mutations and family competition into genetic algorithms as functions optimiser. *Soft Computing* 4: 89–102.
- [YM02] Yoon H. S. y Moon B. R. (2002) An empirical study on the synergy of multiple crossover operators. *IEEE Transactions on Evolutionary Computation* 6(2): 212–223.
- [Zar99] Zar J. (1999) *Biostatistical Analysis*. Prentice Hall.