



TESIS DOCTORAL

**RESOLUCIÓN DE PROBLEMAS DE  
OPTIMIZACIÓN EN COMUNICACIONES  
MÓVILES INALÁMBRICAS MEDIANTE  
COMPUTACIÓN EVOLUTIVA**

**MARÍA DOLORES JARAÍZ SIMÓN**

**DEPARTAMENTO DE TECNOLOGÍAS DE LOS  
COMPUTADORES Y DE LAS COMUNICACIONES**

**2014**



TESIS DOCTORAL

**RESOLUCIÓN DE PROBLEMAS DE  
OPTIMIZACIÓN EN COMUNICACIONES  
MÓVILES INALÁMBRICAS MEDIANTE  
COMPUTACIÓN EVOLUTIVA**

**MARÍA DOLORES JARAÍZ SIMÓN**

**DEPARTAMENTO DE TECNOLOGÍAS DE LOS  
COMPUTADORES Y DE LAS COMUNICACIONES**

Conformidad del Director:

**Fdo: Dr. Juan Antonio Gómez Pulido**

**2014**



*A todos aquellos que se esfuerzan por hacer realidad sus sueños,  
a pesar de las adversidades con las que la vida sorprende.*

*“Hay una fuerza motriz más poderosa que el vapor,  
la electricidad y la energía atómica: la voluntad”.*

**Albert Einstein.**  
Científico alemán, nacionalizado  
estadounidense (1879-1955).



# Agradecimientos

Durante la realización de mis estudios de doctorado he vivido momentos maravillosos y otros difíciles. Pero, en todos ellos, he estado acompañada de gente extraordinaria tanto a nivel personal como a nivel laboral y académico. Personas con las que he compartido confidencias, ilusiones, preocupaciones, emociones, retos y, en definitiva, vivencias.

En primer lugar, quisiera dar las gracias a Juan Antonio Gómez Pulido, mi director de tesis, un excelente profesor y una de las personas más honestas que he tenido el privilegio de conocer. No tengo palabras suficientes para agradecer su dedicación, ayuda y paciencia. Gracias por guiarme en el mundo de la investigación.

Agradezco al grupo de Arquitectura de Computadores y Diseño Lógico de la Universidad de Extremadura por haberme dado la oportunidad de desarrollar esta tesis doctoral. También quiero dar las gracias a los miembros del grupo de Ingeniería de Software de la Universidad de Málaga, a los componentes del grupo de Computación Paralela de la Universidad de La Laguna, y a los integrantes del grupo de Computación Evolutiva y Redes Neuronales de la Universidad Carlos III de Madrid. El hecho de haber colaborado conjuntamente ha sido una experiencia muy enriquecedora para mí.

Me gustaría tener un sentido recuerdo en memoria de Juan Manuel Sánchez Pérez, catedrático de la Universidad de Extremadura en el área de Arquitectura y Tecnología de Computadores. Gracias por sus consejos. Ha sido un honor haberle conocido.

A mis amigos y compañeros de trabajo en la Junta de Extremadura, gracias por su amistad y por los ánimos que siempre me han dado.

Y, por supuesto, quiero agradecer a mi familia, a las personas que son el motor de mi vida, por su cariño, comprensión y apoyo a lo largo de este tiempo. Gracias a una futura personita por hacer que esta dulce espera sea tan especial cada día y por llenarnos de inmensa felicidad.

En estos años he tenido que vivir muy de cerca cómo seres queridos han afrontado la lucha contra el cáncer. Por este motivo, en estas líneas, me gustaría dejar constancia de mi admiración por ellos ya que son un ejemplo de superación. Les doy las gracias porque con su fortaleza y valentía me han enseñado que el sufrimiento puede convertirse en esperanza.

Muchas gracias a todos.

*“Sólo se ve bien con el corazón,  
lo esencial es invisible para los ojos”.*

**Antoine de Saint-Exupéry.**  
Piloto de aviación y escritor  
francés (1900-1944).



# Índice

<b>Resumen</b>	<b>23</b>
<b>Abstract</b>	<b>27</b>
<b>1 Introducción</b>	<b>31</b>
<b>1.1 Motivaciones de la tesis</b>	<b>32</b>
<b>1.2 Evolución y justificación de los trabajos</b>	<b>35</b>
1.2.1 Fase de adquisición de conocimientos	36
1.2.2 Fase de tesis inicial	36
1.2.3 Fase de tesis central	36
1.2.4 Fase de tesis final	37
<b>1.3 Problemas de optimización abordados</b>	<b>38</b>
1.3.1 TS	38
1.3.2 ECC	39
1.3.3 MANET	39
1.3.4 VH	40
<b>2 Computación evolutiva y optimización</b>	<b>43</b>
<b>2.1 Introducción</b>	<b>44</b>
<b>2.2 Paradigmas</b>	<b>45</b>
<b>2.3 Optimización mono-objetivo y multi-objetivo</b>	<b>46</b>
2.3.1 Optimización mono-objetivo	46
2.3.2 Optimización multi-objetivo	46
2.3.2.1 Definición del problema	46
2.3.2.2 Dominancia de Pareto	47
2.3.2.3 Óptimo de Pareto	47
2.3.2.4 Conjunto óptimo de Pareto	47
2.3.2.5 Conjunto no dominado	48
2.3.2.6 Frente de Pareto	48
2.3.3 Problemas de optimización multi-objetivo	48
<b>2.4 Computación Evolutiva</b>	<b>50</b>
2.4.1 Algoritmos Evolutivos	50
2.4.2 Conceptos generales	51
2.4.3 Fundamentos de los algoritmos evolutivos	53
2.4.4 Estructura de un algoritmo evolutivo	54
2.4.5 Tipos de algoritmos evolutivos	56
2.4.6 Ventajas de los algoritmos evolutivos	56
<b>2.5 Algoritmos genéticos</b>	<b>57</b>
2.5.1 Codificación de las soluciones y función de fitness	57
2.5.2 Población, generaciones y diversidad	57
2.5.3 Operadores de un algoritmo genético	58

2.5.4	Funcionamiento de un algoritmo genético	58
<b>2.6</b>	<b>Evolución Diferencial</b>	<b>59</b>
2.6.1	Esquema de Evolución Diferencial	59
2.6.1.1	Mutación	59
2.6.1.2	Cruce	60
2.6.1.3	Selección	60
2.6.2	Estrategias de Evolución Diferencial	62
<b>2.7</b>	<b>Estado del arte en optimización evolutiva multi-objetivo</b>	<b>63</b>
2.7.1	MOGA	63
2.7.2	NSGA	63
2.7.3	NSGA-II	63
2.7.4	PAES	64
2.7.5	SPEA	64
2.7.6	SPEA-2	64
<b>3</b>	<b>Optimización de Series Temporales</b>	<b>65</b>
<b>3.1</b>	<b>Resumen</b>	<b>66</b>
<b>3.2</b>	<b>Introducción</b>	<b>67</b>
<b>3.3</b>	<b>Identificación y predicción de series temporales</b>	<b>68</b>
3.3.1	Modelo ARMAX	70
3.3.2	Algoritmo RLS	70
<b>3.4</b>	<b>Definición del problema de optimización</b>	<b>74</b>
3.4.1	Identificación de sistemas en tiempo real	74
3.4.2	Influencia del forgetting factor en la precisión de la identificación	75
3.4.3	Selección del tamaño óptimo de los modelos	78
3.4.4	Influencia del muestreo de la serie temporal	79
<b>3.5</b>	<b>Benchmarks utilizados</b>	<b>84</b>
<b>3.6</b>	<b>Algoritmo genético paralelo para la optimización de ff</b>	<b>85</b>
3.6.1	Especificaciones del algoritmo	85
3.6.2	Herramientas utilizadas	87
<b>3.7</b>	<b>Resultados experimentales</b>	<b>88</b>
<b>3.8</b>	<b>Conclusiones</b>	<b>90</b>
<b>3.9</b>	<b>Aportaciones</b>	<b>91</b>
<b>4</b>	<b>Optimización del diseño de códigos correctores de errores en comunicaciones</b>	<b>93</b>
<b>4.1</b>	<b>Resumen</b>	<b>94</b>
<b>4.2</b>	<b>Introducción y trabajos relacionados</b>	<b>95</b>
<b>4.3</b>	<b>Códigos correctores de errores</b>	<b>96</b>
<b>4.4</b>	<b>El problema del diseño óptimo de códigos correctores de errores</b>	<b>97</b>
<b>4.5</b>	<b>Aplicación de un AG para la resolución del problema</b>	<b>98</b>
4.5.1	Representación	98
4.5.2	Función de fitness	98
4.5.3	Características del algoritmo genético	99

<b>4.6</b>	<b>Resultados experimentales</b>	<b>101</b>
4.6.1	Herramientas utilizadas	101
4.6.2	Instancias experimentales	101
4.6.3	Tiempos de computación	101
4.6.4	Ejecuciones	102
4.6.5	Resultados con las instancias experimentales	102
4.6.6	Comparativa de rendimiento con otros autores	103
4.6.7	Servicio web de apoyo a la experimentación	107
<b>4.7</b>	<b>Conclusiones</b>	<b>109</b>
<b>4.8</b>	<b>Aportaciones</b>	<b>110</b>
<b>5</b>	<b>Optimización de Redes Móviles Ad-Hoc</b>	<b>111</b>
<b>5.1</b>	<b>Resumen</b>	<b>112</b>
<b>5.2</b>	<b>Introducción</b>	<b>113</b>
5.2.1	Redes inalámbricas Ad Hoc	114
5.2.1.1	Definición	114
5.2.1.2	Clasificación	115
5.2.2	MANET	116
5.2.2.1	Definición	116
5.2.2.2	Características	116
5.2.2.3	Ventajas e inconvenientes	117
5.2.2.4	Clasificación	118
5.2.2.5	Aplicaciones	119
5.2.2.6	VANET	119
5.2.3	Funcionamiento de las redes MANET	119
5.2.3.1	Capa física	119
5.2.3.2	Capa MAC	120
5.2.3.3	Capa de red	120
5.2.3.4	Capa de transporte	122
5.2.3.5	Capa de aplicación	122
<b>5.3</b>	<b>Problema de difusión en MANET metropolitanas</b>	<b>123</b>
5.3.1	Redes móviles ad hoc metropolitanas	124
5.3.2	Escenario del problema	125
5.3.3	Protocolo de broadcasting "Inundación Retardada con Vecindad Acumulativa"	125
5.3.4	Algoritmos evolutivos multi-objetivo	126
5.3.4.1	Codificación	126
5.3.4.2	Función objetivo	127
5.3.4.3	Algoritmo NSGA-II	127
5.3.4.4	Algoritmo evolución diferencial multi-objetivo	129
<b>5.4</b>	<b>Resultados experimentales</b>	<b>130</b>
5.4.1	Parametrización	130
5.4.1.1	NSGA-II	131
5.4.1.2	EDM	131
5.4.2	Indicadores de calidad	131
5.4.3	Experimentos	132
5.4.4	Técnicas paralelas aplicadas a algoritmos evolutivos multi-objetivos	133
5.4.4.1	Características	134
5.4.4.2	Experimentos	134
5.4.5	Comparativa con heurísticas de otros autores	135
5.4.5.1	Proyecto OPLINK	135

5.4.5.2	Análisis comparativo	136
<b>5.5</b>	<b>Conclusiones</b>	<b>138</b>
<b>5.6</b>	<b>Aportaciones</b>	<b>139</b>
<b>6</b>	<b>Optimización de la fase de decisión en el proceso de Vertical Handoff para terminales móviles en redes inalámbricas heterogéneas</b>	<b>141</b>
<b>6.1</b>	<b>Resumen</b>	<b>142</b>
<b>6.2</b>	<b>Introducción</b>	<b>144</b>
6.2.1	¿Qué es Vertical Handoff?	144
6.2.2	Trabajos relacionados	145
6.2.3	Parámetros de calidad de servicio para la fase de decisión	146
6.2.4	Métrica de calidad y función de fitness	147
6.2.5	Parámetros QoS	150
6.2.6	Vertical Handoff en redes de sensores inalámbricos	150
6.2.6.1	Movilidad en redes de sensores inalámbricos	151
6.2.6.2	Conectividad de WSNs heterogéneas basada en calidad de servicio	151
<b>6.3</b>	<b>Problemática de optimización</b>	<b>156</b>
6.3.1	Ajuste de los pesos	156
6.3.2	Características del problema y definiciones	156
<b>6.4</b>	<b>Instancias experimentales</b>	<b>158</b>
6.4.1	Instancia 1	158
6.4.2	Instancia 2	160
6.4.3	Instancia 3	162
6.4.4	Perfiles con las preferencias del usuario	165
<b>6.5</b>	<b>Planteamientos para el problema de optimización</b>	<b>168</b>
6.5.1	Determinación del coste computacional	168
6.5.2	¿Mono-objetivo o multi-objetivo?	170
6.5.3	Algoritmo de optimización	172
<b>6.6</b>	<b>Algoritmo SEFI</b>	<b>173</b>
6.6.1	Características	173
6.6.2	Implementación	174
6.6.3	Resultados experimentales	175
6.6.3.1	Balance precisión/parámetros de calidad para asegurar el tiempo real	175
6.6.3.2	Combinaciones generadas y soluciones del problema	177
6.6.3.3	Obtención de una solución mejor	181
6.6.3.4	Necesidad de una heurística para la selección de una red óptima	181
6.6.3.5	Análisis estadístico de los resultados	184
6.6.4	Conclusiones	184
6.6.5	Claves para la concepción de una nueva heurística	191
<b>6.7</b>	<b>Implementación hardware de SEFI</b>	<b>192</b>
6.7.1	Motivaciones	192
6.7.2	Tecnología y desarrollos	192
6.7.3	Resultados	195
6.7.4	Conclusiones	197
<b>6.8</b>	<b>Algoritmo SEFISA</b>	<b>198</b>
6.8.1	Características	198
6.8.2	Implementación	199
6.8.3	Resultados experimentales	204

6.8.3.1	Tiempo de computación	204
6.8.3.2	Mejora de la solución óptima a lo largo de las generaciones	204
6.8.3.3	Situaciones en las que SEFISA mejora a SEFI	205
6.8.3.4	Análisis de los resultados	210
6.8.4	Comparativa de rendimiento con otros autores	211
6.8.4.1	Estudio de un caso real	211
6.8.4.2	Análisis frente a SEFI	213
6.8.5	Conclusiones	213
<b>6.9</b>	<b>Algoritmo GAVH</b>	<b>214</b>
6.9.1	Características	214
6.9.1.1	Selección	216
6.9.1.2	Cruce	216
6.9.1.3	Mutación	217
6.9.2	Implementación	218
6.9.2.1	Programación del algoritmo	218
6.9.2.2	Instancias experimentales	218
6.9.2.3	Experimentos y ejecuciones	218
6.9.2.4	Estadísticas para los resultados	220
6.9.3	Resultados experimentales	220
6.9.3.1	Barrido paramétrico	220
6.9.3.2	Políticas y porcentaje de mutación.	233
6.9.3.3	Consideraciones estadísticas	234
6.9.4	Comparativa de rendimiento GAVH con SEFISA	234
6.9.5	Rendimiento de GAVH en procesadores empujados	242
6.9.6	Comparativa de rendimiento con otros autores	243
6.9.7	Conclusiones	244
<b>6.10</b>	<b>Conclusiones finales</b>	<b>246</b>
<b>6.11</b>	<b>Aportaciones</b>	<b>247</b>
<b>7</b>	<b>Principales aportaciones y trabajos futuros</b>	<b>249</b>
7.1	Principales aportaciones	250
7.2	Trabajos futuros	251
<b>8</b>	<b>Producción científica</b>	<b>253</b>
8.1	Artículos en revistas internacionales	254
8.2	Capítulos de libro	255
8.3	Comunicaciones en congresos	256
<b>9</b>	<b>Referencias</b>	<b>259</b>



# Índice de figuras

<i>Figura 1.</i>	Problemas de optimización que aparecen cuando un dispositivo móvil se desplaza descubriendo redes inalámbricas heterogéneas, teniendo en cuenta la calidad e integridad de la comunicación, entre otros factores. ....	33
<i>Figura 2.</i>	Planificación y evolución temporal de los trabajos de la tesis doctoral. ....	35
<i>Figura 3.</i>	Frente de Pareto en un problema de optimización con dos funciones objetivo. ....	49
<i>Figura 4.</i>	Estructura genética de individuos en algoritmos evolutivos. ....	53
<i>Figura 5.</i>	Fases y operadores de un algoritmo evolutivo genérico. ....	55
<i>Figura 6.</i>	Pseudocódigo de un Algoritmo Evolutivo. ....	56
<i>Figura 7.</i>	Representación del proceso de cruce para $D = 5$ . ....	60
<i>Figura 8.</i>	Ejemplo bidimensional de una función objetivo mostrando el proceso de generación en el esquema de Evolución Diferencial. ....	61
<i>Figura 9.</i>	Sistema SISO muestreado en tiempo discreto. ....	67
<i>Figura 10.</i>	Tratamiento de la serie temporal. La señal muestreada $y(k)$ es utilizada por un algoritmo de identificación que proporcionará la predicción del comportamiento futuro del sistema. ....	68
<i>Figura 11.</i>	En (a) se muestra una TS, un detalle de la cual es sometida a predicción a largo plazo (b) y a muy corto plazo o a una muestra vista (c), que es realmente lo que supone la IS. ....	69
<i>Figura 12.</i>	Identificación-estimación ARMAX - RLS. La estimación actual $k$ tiene en cuenta las medidas observadas en instantes anteriores: cuantas más se tengan en cuenta, más preciso será el modelo. ( <i>continúa</i> ) ....	71
<i>Figura 13.</i>	El error global en la estimación se basa en los errores anteriores, y es la función <i>fitness</i> que hay que minimizar para una IS más precisa. ....	73
<i>Figura 14.</i>	Influencia de <i>forgetting-factor</i> en la precisión de la identificación. ....	76
<i>Figura 15.</i>	<i>Fitness</i> frente al parámetro <i>forgetting-factor</i> a optimizar. El mínimo de la curva corresponde con el valor óptimo. ....	77
<i>Figura 16.</i>	Diferencia con el $F$ óptimo tal como se indica en la Tabla 3. ....	78
<i>Figura 17.</i>	Valor de la función de <i>fitness</i> $F$ frente a $ff$ , para todos los <i>benchmarks</i> y rangos especificados en la Tabla 3. Obsérvese el mayor o menor pronunciamiento de sus valores mínimos ( $F_{opt}$ ). ....	80
<i>Figura 18.</i>	El periodo de muestreo considerado influye en la diferencia entre la serie muestreada y la serie real. ....	82
<i>Figura 19.</i>	El nº de muestras consideradas influye en la precisión del modelo real. ....	82
<i>Figura 20.</i>	Mejor error cometido en la estimación para 200 valores de $\lambda$ ( $F_{opt}$ ) frente al tamaño del modelo considerado (polinomio ARMAX desde grado 3 hasta 150), $na$ . Análisis hecho para cuatro <i>benchmarks</i> de series temporales de distintos tamaños. ....	83

Figura 21.	Esquema del algoritmo genético paralelo considerado en el problema.....	86
Figura 22.	Interpretación gráfica de un sistema de código corrector de errores.....	97
Figura 23.	Población formada por 7 individuos. El cromosoma de cada individuo está representado por M filas y n columnas.....	98
Figura 24.	Evolución de la solución óptima de las 15 instancias experimentales utilizadas, a lo largo de 150 generaciones. Comparativa hecha según el tamaño del código corrector.....	104
Figura 25.	Evolución de la solución óptima de las 15 instancias experimentales a lo largo de 150 generaciones. La comparativa se ha hecho según el tamaño de las palabras del código corrector.....	105
Figura 26.	Valor del <i>fitness</i> de la solución óptima de las 15 instancias experimentales, para las primeras 4 generaciones del algoritmo genético.....	106
Figura 27.	ECC obtenido por el AG para la instancia experimental considerada.....	107
Figura 28.	Servicio web para la experimentación con AG para el problema ECC.....	108
Figura 29.	Cambio de topología de MANET de 5 nodos, donde se mueve el nodo D.....	118
Figura 30.	Simulador de redes ad-hoc metropolitanas <i>Madhoc</i> .....	124
Figura 31.	Pseudo-Código de NSGA-II.....	128
Figura 32.	Cálculo de la distancia <i>crowding</i> .....	128
Figura 33.	Evolución Diferencial Multi-objetivo basado en el algoritmo NSGA-II.....	130
Figura 34.	Análisis estadístico de los resultados.....	132
Figura 35.	Portal en Internet del proyecto OPLINK: <a href="http://oplink.unex.es">http://oplink.unex.es</a> .....	136
Figura 36.	<i>Vertical Handoff</i> (izquierda) y <i>Horizontal Handoff</i> (derecha).....	144
Figura 37.	Red de sensores inalámbricos heterogénea con tres redes basadas en routers de diferente tecnología, proporcionando distintos valores para los parámetros QoS, y preparada para dar soporte a sensores inalámbricos móviles.....	152
Figura 38.	Un sensor inalámbrico móvil configura un camino donde puede ser alcanzado por la conexión de varios routers de distintas redes heterogéneas. Este esquema se basa en el escenario de la Figura 37. La primera columna indica la posición del sensor móvil en la trayectoria; la segunda informa de cuántos routers disponibles están a su alcance.....	154
Figura 39.	Escenario completo para una red de sensores inalámbricos heterogénea donde pueden darse distintos procesos para VH/HH para el sensor móvil.....	155
Figura 40.	Terminal móvil que, en su desplazamiento dinámico, descubre varias redes inalámbricas con distintas calidades de servicio.....	157
Figura 41.	Instancia experimental número 1. Consiste en 4 redes inalámbricas que proporcionan hasta 10 parámetros de calidad del servicio.....	158
Figura 42.	Instancia experimental número 2. Consiste en 6 redes inalámbricas que proporcionan hasta 4 parámetros de calidad del servicio.....	161
Figura 43.	Escenario correspondiente a la instancia #3, donde existen 6 redes inalámbricas. El terminal móvil se puede desplazar en un área donde hay siempre presentes tres redes de telefonía móvil: dos inalámbricas de área local presentes de forma parcial y una de área local aislada. Los casos A y B de la instancia corresponden a un VH que se produce desde la red inalámbrica UMTS_1 a dos posibles redes inalámbricas	

	de área local, WLAN_1 y WLAN_2, teniendo en cuenta dos servicios de comunicaciones distintos.....	164
Figura 44.	Rangos de pesos para parámetros QoS según los perfiles de usuario.....	167
Figura 45.	Tiempos de computación del algoritmo SEFI para la instancia experimental #1, teniendo en cuenta distintos valores de la precisión en la búsqueda de soluciones para distinto número de parámetros QoS. A mayor precisión, mayor número de combinaciones generadas. El tiempo de computación se emplea tanto para generar las combinaciones como para calcular el <i>fitness</i> de cada una de ellas.....	169
Figura 46.	Número de combinaciones generadas por SEFI (a) y número de éstas que cumplen la restricción del problema de optimización (b), para la instancia experimental #1, teniendo en cuenta distintos valores de la precisión en la búsqueda de soluciones para distinto número de parámetros QoS. ....	171
Figura 47.	SEFI genera todas las posibles combinaciones (vectores de pesos) según una precisión dada h. Se comprueba para cada combinación que se cumpla la restricción y, en caso afirmativo, se calcula el <i>fitness</i> de esa solución para cada red. Las búsquedas estarán siempre acotadas a los límites de los pesos impuestos por los perfiles de usuario. ....	174
Figura 48.	Tiempos de computación de SEFI para la instancia 1 (4 redes inalámbricas), en relación a hasta 6 parámetros QoS, para diversas precisiones de la búsqueda. Se ha señalado en cada gráfica, mediante una zona sombreada, los umbrales de tiempo real en 1 segundo.....	176
Figura 49.	Número de combinaciones generadas y evaluadas por SEFI (gráfica superior de cada página) y número de éstas que satisfacen la restricción del problema (soluciones, gráfica inferior de cada página), frente al número de parámetros QoS, para la instancia experimental #1, y considerando distintos niveles de precisión en la búsqueda. Las gráficas de cada página corresponden a distintos perfiles de usuario. ....	178
Figura 50.	Mejor <i>fitness</i> encontrado por SEFI respecto al número de parámetros de calidad de servicio, para cada precisión. Datos obtenidos con la instancia #1, considerando los tres perfiles de usuario. ....	182
Figura 51.	Tiempo de computación y mejor red encontrada por SEFI, para distinto número de parámetros QoS y precisiones distintas. Cada página de gráficas corresponde con los un perfil de usuario para la instancia #1, mientras que cada gráfica en la página corresponde con una misma precisión. Los círculos señalan datos significativos explicados en el texto. ....	185
Figura 52.	Tiempo de computación y mejor red encontrada por SEFI, para distintos números de parámetros de calidad de servicio considerados, y teniendo en cuenta precisiones distintas. Las gráficas en cada página corresponden a un perfil distinto para la misma instancia experimental #2, mientras que cada gráfica en la página ha sido generada con una precisión distinta. ....	188
Figura 53.	Integración y comportamiento de SEFI en la arquitectura hardware del dispositivo inalámbrico móvil.....	194
Figura 54.	Tarjeta de prototipado Digilent Nexys (izquierda) con FPGA Spartan xc3s500e y Digilent XUPV5 (derecha), con FPGA Virtex5 xc5vlx110t. La primera es una solución económica, de reducido tamaño y bajo consumo, mientras que es una solución de alto rendimiento. ....	195

Figura 55.	Tiempos de computación de SEFI en las implementaciones hardware S3 y V5 para aplicaciones de SEFI que cuentan con perfiles de usuario. La zona sombreada indica lo que se considera como tiempos de respuesta aceptable en entornos de movilidad inalámbrica (1 segundo). ....	196
Figura 56.	Pseudo-código resumido de SEFISA. ....	198
Figura 57.	SEFISA ejecuta SEFI en generaciones sucesivas, donde cada generación ve reducido el espacio de búsqueda de soluciones, dependiendo del factor de reducción, de los límites impuestos por las preferencias de usuario (perfiles) y de los valores óptimos encontrados en la generación anterior, que son utilizados para centrar los nuevos espacios de búsqueda. ....	200
Figura 58.	Caso en el que el espacio de búsqueda para $w_i$ en la generación #J, al reducirse a la mitad en la generación #J+1, excede los límites establecidos por el perfil de usuario e, incluso, el mínimo posible, 0. En casos como éste han de re-ajustarse los límites que son erróneos; esto producirá que, si bien $h$ sigue siendo el mismo, se reduzca el número de posibles valores para generar combinaciones con $w_i$ . ....	202
Figura 59.	Gráfica representativa de la evolución de la solución óptima encontrada a lo largo de las generaciones de SEFISA. Datos obtenidos para la instancia #1 (4 redes inalámbricas de acceso) con 5 parámetros de calidad, para un grado de división de los espacios de búsqueda en SEFISA fijado en 5.....	206
Figura 60.	Rendimiento de SEFISA frente a SEFI, entendido como mejor <i>fitness</i> (solución) encontrado en relación al número total de combinaciones ( <i>total combinations</i> ) y al número de soluciones posibles ( <i>valid combinations</i> ), como gráficas separadas, para la instancia #1 y perfil #1. Cada fila de gráficas corresponde a un número distinto de parámetros de calidad. Los puntos marcados en la línea SEFISA corresponden con los <i>fitness</i> óptimos obtenidos en aquellas ejecuciones con valores del factor de división <i>div</i> iguales a (contando desde el primero a la izquierda) 3, 5, 10, 20, 50 y 100.....	207
Figura 61.	Los mismos gráficos mostrados en la Figura 60, pero para el perfil #2.....	208
Figura 62.	Los mismos gráficos mostrados en la Figura 60, pero para el perfil #3.....	209
Figura 63.	Comparativas de atribuciones de importancia entre los distintos parámetros QoS de la instancia #2 en [Lassoued 2008], según el tipo de aplicación requerida por el usuario (conversación o transmisión multimedia). ....	212
Figura 64.	Selección de la mejor red en cada instante de la trayectoria de un terminal móvil que se desplaza descubriendo distintas redes inalámbricas heterogéneas, teniendo en cuenta los dos posibles tipos de aplicación.....	212
Figura 65.	Planteamiento de la población, individuos y valores de <i>fitness</i> para el algoritmo genético GAVH.....	215
Figura 66.	Operaciones del algoritmo genético que se suceden en una generación.....	216
Figura 67.	Los progenitores $w_A$ y $w_B$ generan el hijo $w_S$ , donde $w_{S_i}$ es el gen o peso QoS $i$ -ésimo del individuo. Este gen se genera de acuerdo con las reglas expresadas en las ecuaciones (42) y (43).....	217
Figura 68.	Mutación por desplazamiento ( $smut=1$ ). Los genes del cromosoma del individuo se desplazan en varias posiciones hacia la derecha (en el dibujo, dos posiciones: $qmut=2$ ). ....	219
Figura 69.	Mutación por intercambio ( $smut=2$ ). Dos genes del cromosoma del individuo intercambian sus posiciones entre sí. Estas posiciones pueden ser fijas en una	

	misma ejecución (pero distintas en otras), en una misma generación (pero distintas en otras), o cambiar aleatoriamente para cada individuo en cada generación y en cada ejecución. ....	219
<i>Figura 70.</i>	Evolución de la mejor solución encontrada por GAVH ( <i>fitness</i> promedio de 50 ejecuciones) respecto al tamaño de la población, para el conjunto de experimentos del barrido paramétrico donde varían <i>NG</i> , <i>popsiz</i> e y <i>smut</i> (con la correspondiente variación de <i>qmut</i> ): (a) $h = 0,005$ y $pmut = 5$ ; (b) $h = 0,01$ y $pmut = 5$ ; (c) $h = 0,005$ y $pmut = 20$ ; (d) $h = 0,01$ y $pmut = 20$ . ....	222
<i>Figura 71.</i>	Extracción de resultados correspondientes a la Figura 70, mostrando solamente los casos que no consideran mutación. ....	225
<i>Figura 72.</i>	Extracción de resultados de la Figura 70, mostrando solamente los casos que consideran mutación $smut=1$ . Importante: en (c) y (d), la dimensión del eje de ordenadas se ha ampliado para mostrar valores inferiores al resto de gráficas, denotando los malos valores obtenidos del <i>fitness</i> . ....	227
<i>Figura 73.</i>	Extracción de resultados correspondientes a la Figura 70, mostrando solamente los casos que consideran mutación $smut=2$ . ....	229
<i>Figura 74.</i>	Desviación estándar de las soluciones óptimas encontradas, para los cuatro casos contemplados en la Figura 70. ....	231
<i>Figura 75.</i>	Influencia del porcentaje de mutación ( $pmut$ ). Para la instancia #1 y para 4 y 3 parámetros QoS, se han evaluado las dos políticas de mutación ( $smut$ ) con distintas posiciones de genes ( $qmut$ ). Las flechas rojas indican los porcentajes óptimos cuando es mejor tener una política de mutación a no tenerla, para el ajuste $\{smut=2; qmut=2\}$ , que es el seleccionado según la Tabla 21, y que son cercanos al valor 5%, decidido en esa misma tabla. ....	235
<i>Figura 76.</i>	Resultados de los mismos experimentos que los de la Figura 75, pero en este caso para la instancia #2. ....	238



# Índice de tablas

Tabla 1.	Relación entre los distintos problemas de optimización abordados en la tesis doctoral, según el marco de ámbitos y objetivos de la misma. ....	38
Tabla 2.	Diferencias entre algoritmo estándar y algoritmo genético. ....	57
Tabla 3.	Para cada <i>benchmark</i> , se compara la función de coste F obtenida para $ff = 0,9896$ , con el F óptimo calculado a partir de un $ff$ encontrado de entre un rango, estableciendo la diferencia entre ambos F en forma de porcentaje. ....	77
Tabla 4.	Principales funciones de la toolbox GADST para operar con AGs. ....	87
Tabla 5.	Comparativa de rendimientos AGP vs. AG, entendiendo por rendimiento el valor mínimo del <i>fitness</i> . Añadimos a la comparativa el porcentaje de cuánto más pequeño es el <i>fitness</i> óptimo del AGP respecto del AGP. ....	88
Tabla 6.	Resultados experimentales del AG (una sola ejecución por experimento). ....	102
Tabla 7.	Comparativa de resultados para la misma instancia experimental. ....	106
Tabla 8.	Valores obtenidos del hipervolumen para los dos algoritmos multi-objetivo. ....	133
Tabla 9.	Valores obtenidos del hipervolumen para los algoritmos paralelos. ....	135
Tabla 10.	Valores obtenidos del hipervolumen para los algoritmos multi-objetivo aplicados en el proyecto OPLINK, ordenados de mejor a peor resultado . ....	137
Tabla 11.	Valores obtenidos del hipervolumen para los algoritmos multi-objetivo paralelos aplicados en el proyecto OPLINK, ordenados de mejor a peor. ....	137
Tabla 12.	Parámetros QoS más habituales. Algunos de estos parámetros deben ser maximizados o minimizados para poder minimizar el <i>fitness</i> (o maximizar la función de mérito); maximizar el parámetro implica mejorar el <i>fitness</i> (a mayor p, menor <i>fitness</i> , tal como se ve en la ecuación (31)), mientras que minimizar el parámetro implica el empeoramiento del <i>fitness</i> (a mayor p, mayor <i>fitness</i> , tal como se ve en la ecuación (30)). ....	153
Tabla 13.	Características de las redes para la instancia #1. ....	159
Tabla 14.	Características de las redes para la instancia #2. La medida del ancho de banda de las distintas redes se obtiene a partir de fuentes referidas por [Lassoued 2008]. Así, para redes IEEE802.11b, el valor está sacado de [Atheros 2001] teniendo en cuenta escenarios realistas, ya que el ancho de banda depende de varios factores, entre ellos la distancia emisor-receptor. La medida del ancho de banda para las redes UMTS es aportada directamente en [Lassoued 2008], mientras que la de las redes WiMax (IEEE 802.16) está sacada de [Betancur 2006]. ....	162
Tabla 15.	Características de las redes para la instancia #3. ....	163
Tabla 16.	Valores de precisión que aseguran, para un número determinado de parámetros QoS, que el tiempo de computación de SEFI permanece por debajo de 1 segundo, umbral definido para mantener el carácter de tiempo real al proceso de selección de redes VH en entornos dinámicos. ....	175

Tabla 17.	Características de las dos implementaciones hardware desarrolladas para el microprocesador empotrado que alberga y ejecuta SEFI. ....	195
Tabla 18.	Valores de precisión ajustados para las implementaciones S3 y V5. ....	196
Tabla 19.	Pesos óptimos asignados a los distintos parámetros QoS para dos tipos de servicio, según la instancia #2 en el trabajo [Lassoued 2008]. ....	212
Tabla 20.	Combinaciones de pesos que satisfacen la restricción (soluciones), para la instancia experimental #2 (6 redes inalámbricas, 5 parámetros QoS). Las soluciones aportadas por [Lassoued 2008] han sido calculadas mediante el algoritmo AHP, para el perfil #2 (conversación, donde importa el retardo D: $0.5 < D < 0.7$ ) y para el perfil #3 ( <i>streaming</i> , donde importa un gran ancho de banda B y un retardo aceptable D: $0.5 < B < 0.7$ , $0.1 < D < 0.3$ ). Las soluciones encontradas por SEFI tienen mucho mejor fitness en todos los casos, aún con baja precisión de búsqueda. ....	213
Tabla 21.	Valores ajustados de los parámetros del algoritmo genético GAVH, para la instancia experimental #1 considerando el perfil de usuario más genérico ( <i>profile=1</i> ), cuatro redes inalámbricas ( $NN=4$ ) y cuatro parámetros de calidad de servicio ( $NQ=4$ )....	233
Tabla 22.	Combinaciones de pesos que satisfacen la restricción (soluciones), para la instancia experimental #2 (6 redes, 5 parámetros QoS). Esta tabla completa la Tabla 20 añadiendo los resultados de GAVH. Las soluciones en [Lassoued 2008] han sido calculadas mediante el algoritmo AHP, para el perfil #2 (conversación, donde importa el retardo D: $0.5 < D < 0.7$ ) y para el perfil #3 ( <i>streaming</i> , donde importa un gran ancho de banda B y un retardo aceptable D: $0.5 < B < 0.7$ , $0.1 < D < 0.3$ ). Las soluciones encontradas por GAVH mejoran sustancialmente los de SEFI, que ya eran buenos respecto a AHP. Por otro lado, el algoritmo GAVH proporciona mejores valores de <i>fitness</i> en todos los casos, aún con baja precisión de búsqueda. ....	241
Tabla 23.	Soluciones (combinaciones de pesos que satisfacen la restricción) generadas por los algoritmos FMADM, GAVH y SEFI, utilizando la misma instancia experimental y perfiles de usuario. ....	244
Tabla 24.	Valores de las funciones de fitness según las formulaciones WNSF y F, correspondientes a las soluciones generadas por los algoritmos FMADM, GAVH y SEFI que se muestran en la Tabla 23 . ....	245

## **Consideraciones para la lectura de esta memoria de tesis.**

- La notación numérica sigue las reglas del idioma español (decimales con coma, millares con punto); sin embargo, en algunas tablas y gráficos de resultados pueden aparecer los números siguiendo la notación inglesa (decimales con punto, millares con coma), dado que se trata de presentaciones utilizadas en trabajos publicados para congresos y revistas internacionales.
- Toda la memoria de la tesis se ha escrito siguiendo un estilo de narración impersonal.



# Resumen

Esta tesis aborda algunos problemas de optimización combinatoria en el campo de las comunicaciones móviles e inalámbricas, y que han sido resueltos mediante la aplicación de algoritmos basados en la computación evolutiva. Estos problemas inciden en diversos escenarios de movilidad para las comunicaciones inalámbricas, especialmente en el ámbito de las redes ad-hoc y en el acceso a las redes de comunicaciones inalámbricas de naturaleza heterogénea de acuerdo con la calidad de servicio.

Por un lado, el desarrollo de la tesis está motivado en la aplicación de la computación evolutiva para resolver problemas de optimización combinatoria, tanto de naturaleza mono-objetivo como multi-objetivo. Con la finalidad de adquirir conocimientos y experiencia que permitiese aplicar satisfactoriamente la computación evolutiva al campo de las comunicaciones móviles inalámbricas, se iniciaron los trabajos de tesis resolviendo un problema de optimización relacionado con la identificación de series temporales, que aportó habilidades útiles para afrontar los trabajos tratados en este documento de tesis.

Por otro lado, la tesis está motivada para resolver tres problemas diferentes de optimización (MANET, ECC, VH) en el ámbito de las comunicaciones inalámbricas móviles. La resolución de estos problemas demuestra que la computación evolutiva es una técnica que proporciona buenos resultados en ambientes estáticos y dinámicos de las comunicaciones inalámbricas.

Si bien estos tres problemas han sido abordados de forma separada, todos ellos están presentes en escenarios comunes. Así por ejemplo, un dispositivo móvil de comunicaciones inalámbricas (tablet, smartphone) que está en movimiento puede establecer comunicación con otros terminales que forman redes de comunicaciones ad-hoc (problema MANET), donde la calidad de la comunicación digital puede ser aumentada mediante una adecuada selección de los códigos correctores de errores (problema ECC). A su vez, este terminal, dentro de la red ad-hoc, puede proporcionar acceso a Internet (para él mismo o para otros terminales) si se conecta a un punto de acceso a Internet. Los puntos de acceso a Internet tienen tecnologías diversas (GPSR, Wifi, 3G, 4G) y diferentes prestaciones para sus principales parámetros de comunicaciones o de calidad del servicio (ancho de banda, retardo, seguridad), con lo que aparece un problema de selección del punto de acceso óptimo (problema VH).

Para abordar estos problemas de optimización se ha seguido una planificación y evolución temporal de los trabajos que son coherentes con el objetivo global de comprobar la efectividad de los algoritmos evolutivos para resolver problemas de optimización en el ámbito de las comunicaciones móviles inalámbricas. Por tanto, en el desarrollo de los trabajos de tesis se pueden identificar cuatro fases: adquisición de conocimientos (problema TS), fase inicial (problema ECC), fase central (problema MANET) y fase final (problema VH).

El problema de **optimización para el aumento de la precisión en la modelación e identificación de las Series Temporales (problema TS)** tiene su ámbito de conocimiento en el tratamiento de la señal digital y en las comunicaciones multimedia, pues la precisión de los modelos matemáticos de las series temporales permite mejorar la predicción de su comportamiento futuro. Esta tesis propone una implementación de Algoritmo Genético Paralelo (AGP) para obtener valores óptimos del parámetro factor de olvido, el cual incide en la precisión de los modelos. La necesidad de un aumento de la precisión de los modelos de las series temporales se justifica principalmente en escenarios donde se requiere una estimación del comportamiento de sistemas críticos (seguridad) y, además, existe una gran variabilidad en sus comportamientos dinámicos (que configuran identificación de escenarios en tiempo real).

El problema de la **optimización de los Códigos Correctores de Errores (problema ECC)** entra en el campo del tratamiento de la información digital transmitida típicamente en las comunicaciones inalámbricas, redes ad-hoc o redes de sensores inalámbricos. En estas comunicaciones, debido al medio aéreo en el que se transmite la información, son habituales los errores debidos a interferencias y ruido, por lo que es necesario establecer mecanismos para detectarlos y, en la medida de lo posible, corregirlos, mediante los ECC. Esta tesis resuelve el problema de obtener un código óptimo mediante algoritmos genéticos, efectuándose numerosos experimentos y ejecuciones para sintonizar sus principales parámetros y políticas. Así, se obtuvieron unos resultados que mejoraron los de algunas otras heurísticas de naturaleza evolutiva.

El problema de **optimización del protocolo de difusión DFCN en redes móviles ad-hoc (problema MANET)** surge cuando se comunican entre sí un gran número de terminales inalámbricos móviles sin infraestructura previa de red y con topología dinámica e impredecible. Realizar la difusión de forma eficiente y fiable en entornos metropolitanos, caracterizados por una alta densidad de nodos heterogéneos y de gran movilidad, es un problema de optimización multi-objetivo donde es necesario ajustar cinco parámetros de este protocolo para satisfacer tres objetivos simultáneos y contrapuestos entre sí: maximizar el número de dispositivos alcanzados (cobertura), minimizar la utilización de la red (ancho de banda) y minimizar la duración del proceso. Esta tesis resuelve este problema de optimización multi-objetivo mediante los algoritmos NSGA-II y Evolución Diferencial Multi-objetivo, utilizando técnicas computacionales paralelas en un marco experimental que utiliza un simulador para modelar una red MANET (escenario de un centro comercial). Los resultados fueron razonablemente buenos al compararlos con los proporcionados por otras heurísticas.

Finalmente, esta tesis resuelve un problema de **optimización de la conexión a puntos de acceso heterogéneos basados en calidad de servicio para terminales móviles inalámbricos (problema VH)**. Este problema, a diferencia de los anteriores, tiene la dificultad añadida de que su resolución está sujeta a bajos tiempos de computación, dada la aplicabilidad en tiempo real que se requiere a las soluciones. Las características de los microprocesadores actuales de bajas prestaciones (comparadas con los existentes en computadores y servidores) que mayoritariamente albergan los dispositivos móviles hacen necesario diseñar

heurísticas de optimización que aporten una buena solución al problema siguiendo dos restricciones: rapidez en la resolución del problema (para satisfacer la utilidad de la aplicación) y ligereza del algoritmo (para satisfacer las bajas prestaciones de memoria y cómputo de los procesadores empujados).

Este problema de optimización centra gran parte de la tesis y genera desarrollos originales y novedosos. Por un lado, el ámbito de estudio contempla escenarios tecnológicos de alto interés hoy en día (redes inalámbricas de tecnología heterogénea -UMTS, WiMax, WLAN-, servicios de usuario -voz, datos-, acceso continuo de alta calidad a la red, entornos dinámicos, calidad de servicio -QoS: ancho de banda, tiempo de respuesta, etc.-), y por otro lado, trata de generar heurísticas inteligentes basadas en la computación evolutiva que sean rápidas, ligeras y precisas, aptas para ser ejecutadas en dispositivos empujados. Esta “inteligencia embebida” introduce un factor diferenciador respecto a los clásicos desarrollos algorítmicos que proporcionan soluciones a los problemas de optimización utilizando potentes sistemas de computación.

El problema de satisfacer la necesidad de ofrecer conexión de alta calidad a terminales móviles mediante puntos de acceso de tecnología heterogénea surge cuando un terminal descubre nuevas redes en su desplazamiento y debe evaluar la calidad del servicio que proporcionan. Decidir a qué nueva red se conecta constituye la fase de decisión del proceso Vertical Handoff (VH), que ha de ser procesado cada vez que se descubre una nueva red o varían los parámetros de calidad, algo frecuente en entornos dinámicos.

Esta tesis resuelve el problema VH proponiendo una novedosa implementación de algoritmo ligero que combina la búsqueda directa con algoritmos evolutivos (enfriamiento simulado y algoritmos genéticos). Los resultados de la implementación basada en el algoritmo genético proporcionan una gran precisión en poco tiempo, mejorando los resultados existentes en la literatura científica.



# ***Abstract***

This thesis tackles some optimization problems that are present in the field of wireless mobile communications. They have been solved by means of algorithms based on evolutionary computation. These problems revolve around different scenarios of mobility for wireless communications, especially in the fields of ad-hoc networks and connectivity to heterogeneous wireless access networks according to the quality of service.

On the one hand, the development of the thesis is motivated in the application of the evolutionary computation for solving combinatorial optimization problems of mono-objective and multi-objective nature. In order to achieve knowledge and experience to apply successfully the evolutionary computation to the wireless mobile communications field, the first step of the research was to solve an optimization problem related with the identification of time series; therefore, this problem provided useful skills to be used in further problems in the thesis document.

On the other hand, the thesis is clearly oriented to solve three different optimization problems (MANET, ECC, and VH) that are present in the wireless mobile communications. After solving these problems, the thesis shows that evolutionary computation is a useful technique to obtain good results in static and dynamic environments of the wireless communications.

Although these problems have been tackled by separated ways, all they are present in common scenarios. Thus, for example, a wireless device (tablet, smartphone) moving along a determined scenario can establish communication with other terminals building ad-hoc networks (MANET problem) where the digital communication quality can be increased by means of an adequate selection of the error correcting codes (ECC problem). In turn, this terminal, within the ad-hoc network, can give access to Internet (to itself or to other terminals) if it connects to an access point. The access points have diverse technologies (GPSR, Wi-Fi, 3G, 4G) and different performances for their main parameters of quality of service (bandwidth, delay, security level, cost). As a result, an optimization problem arises for selecting the best access point (VH problem)

In order to study these optimization problems, the thesis has followed a planning and evolution of tasks which are coherent with the global objective to prove the effectiveness of the evolutionary algorithms to solve optimization problems in the field of wireless mobile communications. In this way, the research and development tasks of the thesis have been grouped in four consecutive phases: knowledge acquisition (TS problem), initial phase (ECC problem), central phase (MANET problem) and final phase (VH problem).

The optimization problem for improving the precision in the modelling and identification of time series (TS problem) has its knowledge field in the digital

signal processing and in the multimedia communications, since the precision of the mathematical models of the time series allows improving the prediction of the future behaviour. This thesis suggests a Parallel Genetic Algorithm (AGP) in order to obtain optimal values of the forgetting factor, which is the most important parameter for determining the precision of the models. The need for improving the precision of the time series model is justified mainly by scenarios where an estimate of the behaviour of critical systems (safety) is required; and furthermore, there is a great variability in their dynamic behaviours (which configure scenarios of identification in real time).

The error correcting code optimization (ECC problem) belongs to the digital communications field, where the information is transmitted typically by means of wireless, ad-hoc, or sensor networks. In these communication systems, and due to the aerial environment, it is usual to produce errors in the digital information because of the noise and interferences. For this reason it is necessary to establish mechanisms for detecting and, if it is possible, and correcting the errors by means of the error correcting codes. This thesis solves the problem of obtaining the optimal code using genetic algorithms. For this purpose, a lot of experiments and executions have been done to select the main parameters and strategies of the genetic algorithm. This way, good results were obtained, improving the solutions produced by other heuristics of evolutionary nature.

The problem to find the optimal parameters of the DFCN broadcasting protocol in ad-hoc wireless networks (MANET problem) arises when a big number of wireless mobile terminals compose a network without previous infrastructure and unpredictable topology. An efficient and reliable broadcasting in metropolitan environments, characterized by a high density of heterogeneous moving nodes, is a multi-objective optimization problem where five parameters of the broadcasting protocol need to be adjusted in order to satisfy three simultaneous and contradictory objectives: maximizing the number of achieved devices (coverage), minimizing the network use (bandwidth) and minimizing the process time. This thesis solves this multi-objective optimization problem by means of NSGA-II and Differential Evolution algorithms, using parallel computational techniques in an experimental framework that uses a simulator to model a metropolitan ad-hoc network (specifically, a shopping center). The results obtained by these algorithms were good in comparison with the obtained by other heuristics from third authors.

Finally, and as the most important and complex problem tackled, this thesis solves an optimization problem to choose the best access point among different heterogeneous wireless networks based on the quality of service, in order to give access to a mobile terminal (VH problem), according to the vertical handoff process. This problem stands out the rest of the problems tackled in the thesis in an added difficulty: finding the solution is subjected to low computing time, because of the application of the optimization algorithm to the real microprocessors that are present in the current mobile terminals. These low performance processors (in comparison with the existing ones in servers and computers) force us to design optimization heuristics able to supply a good solution of the problem observing two constraints: a quick solving of the problem

and a light algorithm, in order to satisfy the low memory and computing resources of the embedded microprocessors present in the mobile devices.

This optimization problem constitutes the main effort of the thesis and generates original and novel developments and proposals. On the one hand, the application field is composed of a high-interest technological environment nowadays: heterogeneous wireless access networks (UMTS, WiMax, and WLAN), user services (voice, data), steady high-quality network access, dynamic scenarios, quality of service (bandwidth, delay, jitter, cost, and security, among other quality parameters), etc. On the other hand, the thesis proposes light, precise and quick intelligent heuristics based on evolutionary computation, able to be computed on embedded low power and low performance microprocessors present in current mobile devices. This "embedded intelligence" introduces a different aspect with regard to the classic algorithms used to solve combinatory optimization problems that need powerful computing systems.

The problem of satisfying the need for giving high-quality connection to mobile terminals by means of access points of heterogeneous technologies arises when a terminal discovers new networks along its trajectory, and must evaluate the quality of service that they supply. The decision phase of the Vertical Handoff process consists of deciding which network is the best to connect to, and this phase must be processed each time a new network is discovered, or if any of the network quality parameter changes (an usual issue in dynamic environments).

This thesis solves the VH problem suggesting a novel implementation of a light algorithm that combines the direct search with evolutionary algorithms (simulated annealing and genetic algorithms). The results obtained from the genetic algorithm proposed in this thesis supply a high precision in low time, improving existing results from algorithms developed by other authors.



# ***1 Introducción***

*“Esta máquina puede hacer  
cualquier cosa que sepamos  
cómo ordenarle que la ejecute.”*

**Ada Augusta Byron**  
también llamada Ada Lovelace.  
Matemática y escritora inglesa.  
Reconocida como la primera  
programadora de la historia (1815-1852).

## 1.1 Motivaciones de la tesis

El cada vez más universalizado acceso a Internet ha propiciado la aparición de un gran número de dispositivos, tecnologías y servicios al alcance del usuario para establecer comunicaciones en cualquier lugar y situación. En este ámbito, destaca fuertemente la componente de **movilidad en las comunicaciones inalámbricas**, materializado por los populares dispositivos móviles, tan presentes y cotidianos hoy en día (smartphones, tablets).

En este ámbito específico de movilidad en las comunicaciones inalámbricas se presentan diversos problemas de optimización combinatoria que necesitan de la aplicación de técnicas de computación evolutiva para su resolución.

Los problemas de optimización abordados en esta tesis pertenecen a un ámbito de investigación de **gran importancia para el futuro de Internet**. Así, esta tesis insistirá mucho, a través de los principales problemas estudiados (MANET y VH), en los escenarios de **movilidad** para las comunicaciones inalámbricas.

La movilidad (en redes ad-hoc y en comunicaciones inalámbricas heterogéneas), es uno de los cinco intereses de investigación claves de la llamada “Internet del Futuro” o “Internet 3.0” [Pan 2011]<sup>1</sup>. Además, esta tesis aborda un problema de optimización (ECC) relacionado con la seguridad en la transmisión digital de las comunicaciones inalámbricas, a través de la incorruptibilidad de la información transmitida, que es otro de esos cinco intereses claves de Internet 3.0.

Teniendo en cuenta el ámbito de las comunicaciones móviles, las principales motivaciones de esta tesis se resumen en los siguientes puntos:

- Computación evolutiva. Se pretendió adquirir experiencia en la aplicación de la computación evolutiva para resolver problemas de optimización, de naturaleza mono-objetivo y multi-objetivo. Para ello, se decidió inicialmente resolver un problema de optimización relacionado con la identificación de series temporales, como base para acometer los problemas centrales de la tesis.
- Comunicaciones móviles inalámbricas. Se fijó como objetivo la resolución de algunos problemas de optimización presentes en el ámbito de las comunicaciones inalámbricas móviles. Así, **se decidió abordar tres problemas diferentes con el objetivo de abrir el abanico de soluciones en este mismo ámbito, así como para verificar la idoneidad de aplicar la computación evolutiva como técnica eficiente de resolución de problemas que se presentan cuando se establecen comunicaciones en redes inalámbricas en ambientes estáticos y dinámicos.**

En la Figura 1 pueden observarse algunos problemas que aparecen en el ámbito de las comunicaciones móviles inalámbricas, concretamente cuando un dispositivo que se está moviendo y que dispone de capacidad de comunicación inalámbrica

---

<sup>1</sup> “*Mobility and ubiquitous access to networks: The Internet is experiencing a significant shift from PC-based computing to mobile computing. Mobility has become the key driver for the future Internet. Convergence demands are increasing among heterogeneous networks such as [...] wireless ad hoc or sensor networks [...]. Putting mobility as the norm instead of an exception of the architecture nurtures future Internet architecture with innovative scenarios and applications*”.

realiza una conexión (GPSR, Wi-Fi, etc.) con otro terminal, bien con una antena o estación base de transmisión que, a su vez, puede ser un punto de acceso que le permite conexión a Internet.

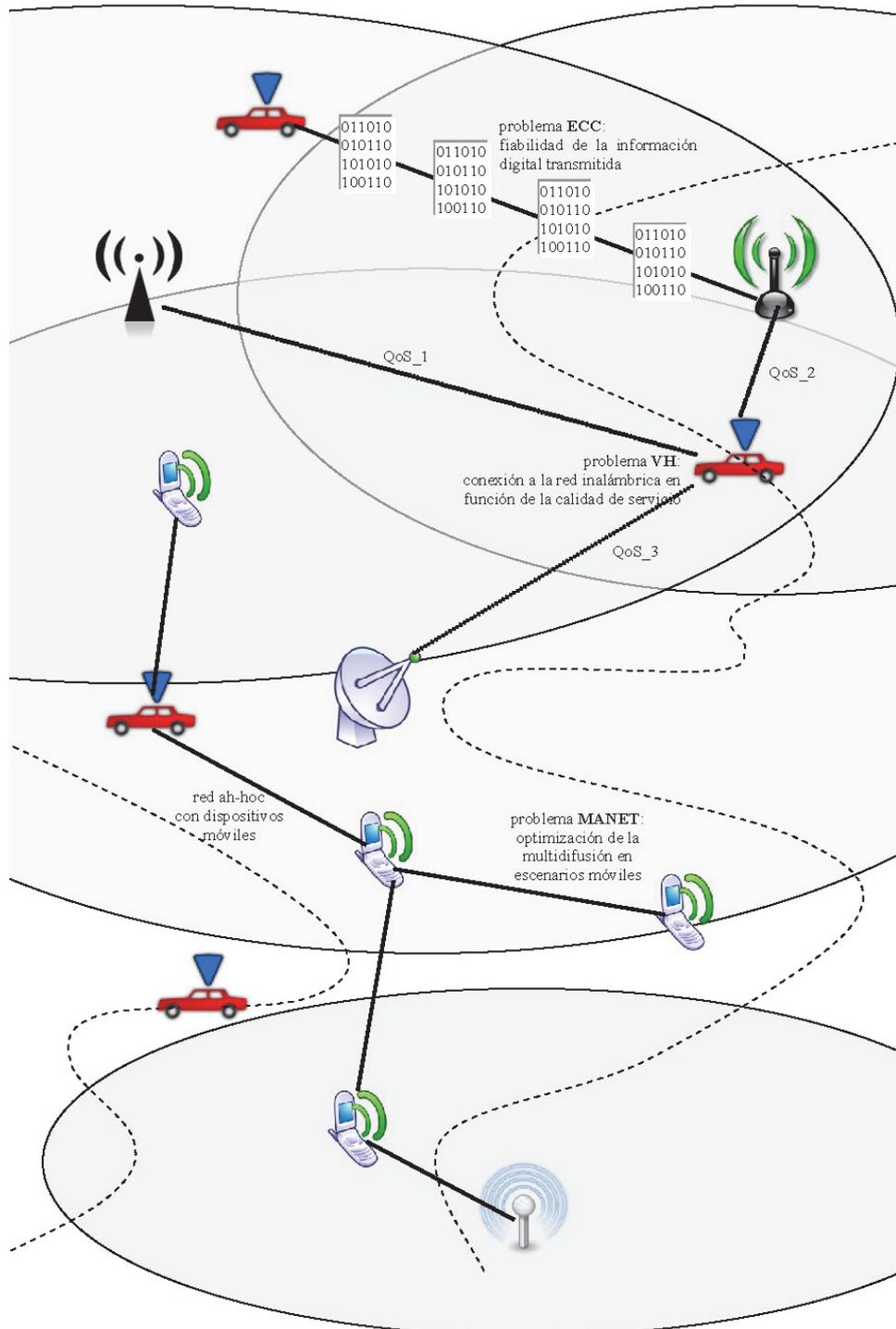


Figura 1. Problemas de optimización que aparecen cuando un dispositivo móvil se desplaza descubriendo redes inalámbricas heterogéneas, teniendo en cuenta la calidad e integridad de la comunicación, entre otros factores.

El primer problema estudiado en este ámbito tiene relación con la integridad de la señal transmitida por vía inalámbrica. Cuando la comunicación se desarrolla por esta vía, es mucho más común sufrir distorsiones e interferencias que cuando se utiliza una vía cableada. Por ejemplo, los factores medioambientales (temperatura, humedad, viento, presión, luz) y geográficos (obstáculos), entre otros factores (como la presencia de instalaciones eléctricas que producen interferencias), condicionan enormemente la integridad de la comunicación digital, especialmente cuando la conexión se establece en un entorno de movilidad, que hace que dichos factores cambien continuamente y con intensidad variable. Debido a estos condicionantes del entorno, los microprocesadores que gobiernan los terminales móviles deben comprobar la calidad de la señal según protocolos establecidos donde unos códigos correctores de error minimizan la pérdida o corrupción de la información transmitida. Así, en este ámbito surge el problema de determinar la calidad de dichos códigos correctores de error, para lo cual se utilizan unas funciones de mérito o fitness que determinan cuál, de las muchas combinaciones posibles de códigos correctores existentes, proporciona la calidad óptima.

Siguiendo con este escenario, supóngase que el terminal móvil establece una conexión inalámbrica con otro dispositivo durante un tiempo limitado, tras el cual establece conexión un tercero, y así de forma continua, conformando una red ad-hoc de topología impredecible, debido al alto grado de movilidad de los terminales. Esta situación es típica en ambientes metropolitanos (p.e., en centros comerciales), donde la alta densidad de dispositivos permite establecer múltiples posibilidades de conexión. En estas redes ad-hoc es común encontrar un protocolo de difusión o *broadcasting* que es susceptible de ser optimizado en sus principales parámetros, con el objeto de proporcionar la máxima calidad a la comunicación en la red. En este caso se estudia la optimización de múltiples parámetros de forma simultánea, lo cual genera un problema de optimización multi-objetivo.

Por último, el terminal móvil puede establecer, en este entorno de movilidad, una conexión inalámbrica con un punto de acceso a Internet mediante el descubrimiento de diversas redes de acceso con tecnologías heterogéneas. Si bien cualquiera de estas redes puede proporcionarle acceso en un momento dado, no todas ellas se lo proporcionan con la misma calidad de servicio, ya que cada una de ellas dispone de valores distintos para sus principales parámetros, como pueden ser el ancho de banda, el retardo, el coste del servicio, etc.

Evidentemente, los parámetros de calidad de servicio cambian para cada red, por lo que, según el servicio de comunicación deseado o el perfil del usuario, es crucial realizar una selección óptima de la red con la que establecer la conexión. De esta manera surge un problema de optimización combinatoria de naturaleza mono-objetivo. Además, las redes de conexión inalámbricas cambian durante el tiempo para el terminal en este escenario dinámico (porque el terminal se mueve descubriendo nuevas redes, acercándose o alejándose de otras viendo afectados sus parámetros de calidad), por lo que el problema de optimización ha de ser resuelto en un instante dado, para satisfacer la experiencia del usuario. Esto fuerza a buscar alternativas algorítmicas evolutivas muy rápidas y ligeras que ofrezcan soluciones en tiempos muy pequeños sin mermar las prestaciones de los microprocesadores disponibles en los dispositivos móviles inalámbricos.

## 1.2 Evolución y justificación de los trabajos

Tal como se ha indicado anteriormente, en el apartado de motivaciones de la tesis, los trabajos desarrollados tratan de cubrir varios objetivos parciales que configuran un objetivo global: probar la efectividad de los algoritmos evolutivos para resolver problemas de optimización en el ámbito de las comunicaciones móviles inalámbricas. Para ello, se han desarrollado los trabajos siguiendo una planificación y evolución temporal coherente con dicho objetivo último.

En el desarrollo de estos trabajos de tesis se pueden identificar cuatro fases, tal como puede observarse en la Figura 2. Estas fases se suceden en un apoyo consecutivo de conocimientos y en una evolución de objetivos parciales para llegar al objetivo final.

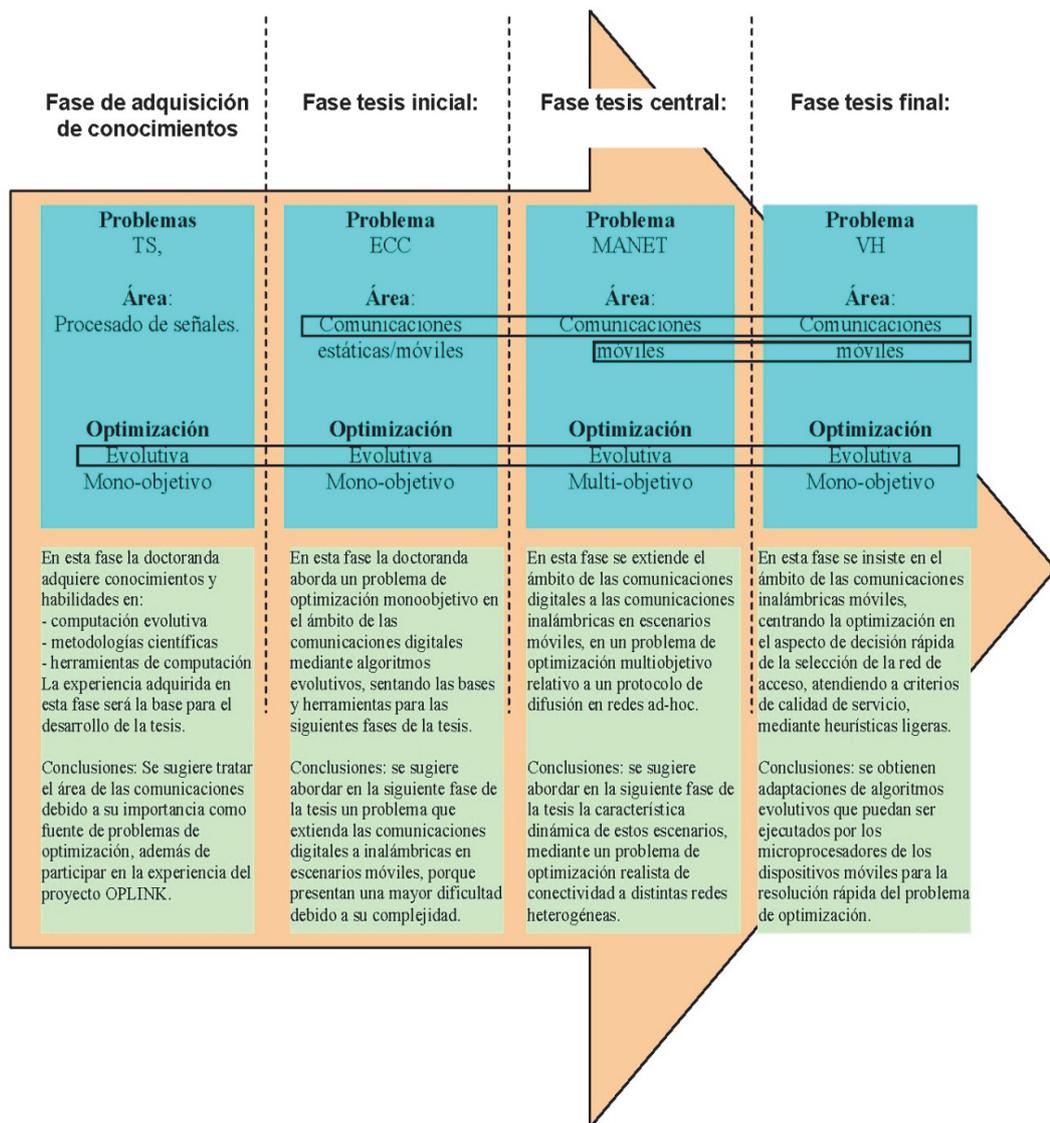


Figura 2. Planificación y evolución temporal de los trabajos de la tesis doctoral.

### **1.2.1 Fase de adquisición de conocimientos**

Esta fase no desarrolla trabajos propiamente dichos de la tesis doctoral, sino tareas de investigación que aportan a la doctoranda conocimientos, habilidades y experiencias que han sido aplicadas en los distintos trabajos de la tesis.

De esta forma, se podría considerar a esta fase como una etapa necesaria para permitir iniciar la línea troncal de la tesis doctoral, ya que diversos aspectos de esta fase se aplicaron posteriormente.

Esta fase consistió en la resolución de un problema de optimización (problema TS) sobre la identificación de series temporales utilizando algoritmos genéticos. Los conocimientos sobre algoritmos genéticos fueron aplicados posteriormente a los problemas ECC (algoritmo genético clásico), MANET (algoritmo genético multi-objetivo NSGA-II) y VH (algoritmo genético empotrado de alta velocidad).

### **1.2.2 Fase de tesis inicial**

La siguiente fase de tesis se puede considerar como la fase inicial de la misma, en cuanto a que se comienza a abordar el área de las comunicaciones móviles inalámbricas atendiendo al problema de la integridad de la comunicación en entornos de ruido y alteración típicos en la transmisión aérea (vía inalámbrica), donde se producen errores de transmisión que deben ser corregidos y minimizados. Con este propósito, se aplicaron técnicas evolutivas en un enfoque mono-objetivo.

Al concluir este trabajo se pudo constatar la conveniencia de añadir el ámbito de la movilidad en la misma comunicación inalámbrica, como ocurre en las redes ad-hoc metropolitanas, añadiendo así mayor realismo a la investigación.

### **1.2.3 Fase de tesis central**

Los escenarios de redes ad-hoc metropolitanas ofrecen una mayor dificultad que el problema anterior, debido a su complejidad, lo cual derivó también en un enfoque de optimización multi-objetivo. Esta fase puede ser considerada como central en cuanto a que se abordan las técnicas y ámbitos nucleares de la tesis: comunicaciones inalámbricas, movilidad, optimización y algoritmos evolutivos.

El problema de optimización abordado (MANET) es un problema multi-objetivo relativo al protocolo de difusión en redes ad-hoc. Al concluir la investigación en este problema, y siguiendo una progresión coherente en la tesis, se consideró oportuno abordar una última fase de investigación que abordase la característica dinámica de las comunicaciones inalámbricas.

Dado que la movilidad de los terminales produce importantes problemas de conectividad a redes heterogéneas que hay que optimizar, atendiendo a criterios de calidad de servicio e inmediatez en la proporción de las soluciones, esta fase aporta metodologías y técnicas para resolver estos problemas. Abordar este tipo de escenarios aumenta aún más el carácter práctico de esta investigación.

#### **1.2.4 Fase de tesis final**

En esta fase final de la tesis desembocan, como se acaba de indicar, los ámbitos de investigación anteriores, al tratar un problema de optimización de las comunicaciones móviles inalámbricas, esta vez enfocadas a la conexión a redes heterogéneas, que son las que proporcionan los servicios de acceso a voz y datos a los terminales móviles. En este problema se atienden aspectos de mucho interés hoy día, como es la calidad de servicio.

Al tratarse de un problema de optimización resuelto con algoritmos evolutivos, éstos han de adaptarse a las restricciones hardware (microprocesadores de bajas prestaciones) y software (memoria disponible limitada) de los terminales móviles más habituales (smartphones, tabletas) que hagan su aplicación realista, lo cual ha requerido un gran esfuerzo en el diseño de soluciones software de optimización rápidas y de buen rendimiento.

De esta forma, se realizaron adaptaciones de algoritmos evolutivos clásicos (algoritmos genéticos y de enfriamiento simulado) que pueden ser ejecutados bajo estas restricciones, como un enfoque de inteligencia computacional empotrada.

### 1.3 Problemas de optimización abordados

A continuación se introducen brevemente los distintos problemas de optimización abordados en la tesis doctoral, y que se explicarán con detalle en los próximos capítulos, a fin de encuadrar su relación mutua en cuanto al enfoque, ámbito y objetivos de la tesis, tal como se observa en la Tabla 1.

Problema	TS	ECC	MANET	VH
	Precisión en la identificación de series temporales	Detección de códigos correctores de errores en comunicaciones inalámbricas	Difusión de la comunicación entre terminales móviles en redes inalámbricas ad-hoc	Conexión a redes inalámbricas heterogéneas según calidad de servicio en dispositivos móviles
Optimización	Mono-objetivo	Mono-objetivo	Multi-objetivo	Mono-objetivo
Comunicaciones inalámbricas	No	Sí	Sí	Sí
Movilidad	No	No	Sí	Sí
Utilización de alg. genéticos	Sí	Sí	Sí	Sí

Tabla 1. Relación entre los distintos problemas de optimización abordados en la tesis doctoral, según el marco de ámbitos y objetivos de la misma.

#### 1.3.1 TS

Resolver el problema del aumento de la precisión en la modelación e identificación de las Series Temporales (TS) tiene su interés en el campo del tratamiento de la señal y en las comunicaciones multimedia, puesto que una mejor modelación de las ecuaciones matemáticas que describen las series temporales aumentan la precisión de la predicción de su comportamiento futuro. Con este propósito, se ha propuesto en esta tesis doctoral un modelo de aproximación a un algoritmo genético paralelo, simulado en un entorno monoprocesador, cuyos resultados se han comparado con los obtenidos en un AG simple, utilizando 12 benchmarks de series temporales de distintos tamaños.

Para desarrollar este trabajo se han empleado técnicas de Identificación de Sistemas (IS) mediante el modelo *Autoregressive Moving Average Extensive* (ARMAX) y el algoritmo recursivo de mínimos cuadrados *Recursive Least Squares* (RLS). En este algoritmo existe un parámetro (*forgetting-factor*) cuyo valor es susceptible de ser optimizado (sintonizado o ajustado), ya que hay muchas series temporales cuya identificación es muy sensible a este valor.

Esta optimización es importante porque influye en la precisión de la identificación y, a partir de ella, en una mejor predicción del comportamiento futuro de la serie temporal. Esta precisión se justifica cuando se tratan problemas cuya naturaleza exige de un mínimo error en la estimación (sistemas críticos), o cuando tienen una gran variabilidad en su comportamiento dinámico (identificación en tiempo real).

La función objetivo que se pretende minimizar depende del valor de dicho parámetro, y representa el error en la estimación global de la serie temporal.

### **1.3.2 ECC**

Es típico que, en las comunicaciones inalámbricas, redes ad-hoc o redes de sensores inalámbricos, la transmisión de la información digital sufra interferencias y ruido por factores del entorno que produzca errores, por lo que es necesario establecer mecanismos para detectarlos y, en la medida de lo posible, corregirlos. Esto es importante porque la repetición de un mensaje corrompido minora el rendimiento y aumenta el coste de la comunicación.

La alternativa a la repetición del mensaje consiste en corregirlo en el destino, utilizando los llamados Códigos Correctores de Errores (ECC), que añaden redundancia al mensaje para controlar el error, lo cual permite detectar y/o corregir los errores ocurridos durante la transmisión de los bloques de datos.

El problema de optimización abordado en esta tesis consiste en encontrar el código óptimo, de entre el muy elevado número de posibles códigos correctores, dado un número fijo de palabras en el código y un tamaño de palabra también determinado. La resolución de este problema fue abordada mediante algoritmos genéticos, para lo cual se efectuaron numerosos experimentos y ejecuciones para sintonizar los parámetros y las políticas de los operadores del algoritmo genético. De esta manera se obtuvieron unos resultados que mejoraron los de algunas otras heurísticas de naturaleza evolutiva.

### **1.3.3 MANET**

Las redes MANET surgen al comunicarse entre sí un gran número de terminales inalámbricos móviles sin infraestructura previa para la comunicación. La movilidad y el alcance limitado de las conexiones inalámbricas hacen impredecible la topología de la red, que es cambiante en el tiempo.

El problema de optimización surge al utilizar un protocolo de difusión (DFCN) mediante el cual un terminal envía un mensaje al resto. Realizar la difusión de forma eficiente y fiable es un reto en este entorno, especialmente en escenarios metropolitanos o urbanos, caracterizados por una alta densidad de nodos heterogéneos y de gran movilidad.

Por otro lado, el ajuste de cinco parámetros de este protocolo configura un problema de optimización multi-objetivo, ya que se busca satisfacer tres objetivos simultáneos y contrapuestos entre sí:

- Maximizar el número de dispositivos alcanzados (cobertura)
- Minimizar la utilización de la red (ancho de banda)
- Minimizar la duración del proceso.

El ámbito experimental se ha llevado a cabo mediante un simulador que modeló una red MANET implementando un escenario de centro comercial.

Para la resolución de este problema de optimización multiobjetivo se utilizaron los algoritmos NSGA-II y Evolución Diferencial Multi-objetivo, y se implementaron técnicas computacionales paralelas con el propósito de afrontar alto coste computacional al considerar un alto número de evaluaciones.

Los resultados fueron razonablemente buenos al compararlos con los proporcionados por otras heurísticas.

#### **1.3.4 VH**

La fase final de la tesis aborda un problema de optimización en el ámbito de las comunicaciones móviles sobre redes inalámbricas heterogéneas, donde se considera la calidad del servicio como la principal característica a considerar a la hora de establecer una conexión a una red que, habitualmente, proporciona acceso a Internet (punto de acceso).

Resolver este problema de optimización de forma rápida y con un bajo coste computacional es indispensable debido a la restricción impuesta por los microprocesadores de bajas prestaciones que mayoritariamente albergan muchos de los dispositivos móviles que establecen las conexiones. Por tanto, la consecución de heurísticas que aporten una buena solución al problema tienen que ajustarse obligatoriamente a estas dos restricciones: rapidez en suministrar la solución (cercanía al tiempo real) y ligereza del algoritmo (para ser ejecutado en microprocesadores empotrados con escasa capacidad de memoria de programa).

El problema abordado en esta tesis tratará con redes inalámbricas de tecnología heterogénea (UMTS, WiMax, WLAN, etc.), capaces de proporcionar diversos servicios (voz, datos), y que se considerarán como puntos de acceso a la red para terminales que necesitan mantener la conectividad en todo momento. Estas redes heterogéneas son utilizadas como puntos de acceso a la red para el terminal móvil.

Así, este es un entorno dinámico donde hay que tener en cuenta la movilidad del terminal a la hora de desarrollar algoritmos capaces de proporcionar soluciones óptimas rápidamente. Esta es una característica diferenciadora frente a otros problemas de optimización en el campo de las comunicaciones inalámbricas, como son el posicionamiento de nodos, la arquitectura de redes, la asignación de bandas de frecuencia, problemas que permiten la ejecución de algoritmos que obtienen soluciones óptimas al cabo de muchas horas.

Cada red está caracterizada por una serie de parámetros que proporcionan la llamada calidad del servicio (QoS), como el ancho de banda, tiempo de respuesta, etc. Sus valores indican la calidad de una red en cuanto a un servicio determinado.

Este problema de optimización surge a partir de la necesidad de ofrecer conexión ininterrumpida de máxima calidad a terminales móviles que pueden desplazarse a gran velocidad en escenarios donde están disponibles varias redes de distinta tecnología. De esta forma, cuando el terminal descubre nuevas redes de acceso en su desplazamiento, ha de evaluar la calidad del servicio que proporcionan con el fin de tomar la decisión de si abandona la actual conexión o no, y en caso afirmativo, decidir a qué nueva red conectarse.

Este proceso se denomina Vertical Handoff (VH), y se compone de tres etapas o fases, la más importante de las cuales es la fase de decisión. La fase de decisión VH ha de ser ejecutada cada vez que se descubre una nueva red (o varían los parámetros de la actual), cosa frecuente en entornos dinámicos, de ahí la necesidad de encontrar algoritmos de decisión rápidos.

Para una red dada, su conjunto de parámetros QoS posee ciertos valores; el peso normalizado que se le asigne a cada uno de ellos permite obtener una medida global de calidad de la red, constituyendo una posible solución. De entre todas las posibles soluciones (combinaciones de pesos que cumplen ciertas restricciones), la óptima será aquella que ofrezca un mejor valor de la medida de calidad de esa red. Si se evalúa este proceso de optimización para todas las redes que el terminal puede ver, se podrá determinar cuál es la solución óptima global (la mejor de entre las mejores soluciones de todas las redes).

Por tanto, este problema de optimización consiste en, para cada red inalámbrica, realizar una búsqueda de la solución óptima basada en calidad de servicio de entre todas las existentes en el espacio de soluciones y, una vez hecho, determinar la mejor de las soluciones óptimas para escoger la red a la que conectarse. Ante posibles espacios de soluciones muy grandes (que aumentan al contemplar un mayor número de parámetros QoS) y tener que obtenerse soluciones óptimas en tiempo real, las búsquedas directas están claramente descartadas, debiéndose recurrir a heurísticas rápidas de búsqueda no exhaustiva.

Es importante tener en cuenta que la mayoría de los terminales móviles ejecutarían los algoritmos de decisión VH mediante microprocesadores de no muy altas prestaciones, dadas las necesidades de bajo consumo y alto grado de integración, por lo que el desafío de abordar este problema de optimización aumenta al ser necesario implementar heurísticas ligeras y rápidas.

En esta tesis se ha desarrollado una propuesta novedosa de algoritmo de búsqueda directa (que evalúa todas las posibilidades) no exhaustiva (para sub-espacios de los espacios posibles de soluciones) que tiene un coste computacional proporcional a la precisión requerida a la búsqueda. Este algoritmo ha sido utilizado para integrarse en dos propuestas de heurísticas evolutivas, basadas en el enfriamiento simulado y en algoritmos genéticos, evitando elevar el tiempo de computación, cosa que no puede evitar dicho algoritmo.



# ***2 Computación evolutiva y optimización***

*“La práctica debe estar  
siempre basada en el buen  
conocimiento de la teoría.”*

**Leonardo da Vinci.**

Pintor, ingeniero, arquitecto, escultor,  
inventor, artista, científico, filósofo,  
escritor, anatomista, botánico, poeta,  
músico y urbanista (1452-1519).

## 2.1 Introducción

El principal objetivo de cualquier técnica de optimización es encontrar el óptimo global, es decir, la mejor solución posible para un problema determinado. Sin embargo, no se puede garantizar que en todos los problemas de optimización se consiga una convergencia a la solución óptima.

Los problemas cuya complejidad está acotada por un polinomio son los denominados **problemas P** (polinómicos). Así, se dice que un problema de optimización es un problema polinómico o P, si puede ser resuelto en tiempo polinomial por una máquina de *Turing* determinística. Esto significa que el algoritmo se comporta de manera predecible, es decir, que no tiene más que un único camino posible a seguir.

En el ámbito teórico de la complejidad computacional, un **problema NP** (acrónimo de *Non-Deterministic Polynomial-time* en inglés, que significa *Polinómico No Determinista*) es aquél que puede ser resuelto en tiempo polinómico por una máquina de Turing no determinista. También existen los **problemas NP-completos**, en los que los algoritmos que los resuelven emplean un tiempo exponencial, en el peor de los casos.

La presente tesis resuelve alguno de estos problemas, de naturaleza combinatoria, donde el elevado número de posibles soluciones hace necesario recurrir a heurísticas de optimización que no están basadas en técnicas exactas.

Por otro lado, los problemas abordados en esta tesis se pueden clasificar en dos tipos, que implican técnicas de resolución y métricas de efectividad completamente diferentes:

- **Problemas de optimización mono-objetivo.** Son problemas donde sólo existe un objetivo a optimizar, y se abordan mediante la minimización o maximización de su función de *fitness* correspondiente. Las heurísticas que los resuelven suelen hacer converger las soluciones, a lo largo de las sucesivas generaciones, a la solución óptima.
- **Problemas de optimización multi-objetivo.** Son problemas donde existe más de un objetivo a optimizar, mediante la minimización o maximización de sus correspondientes funciones de *fitness*, a menudo contrapuestas. Estos problemas presentan soluciones que son conjuntos de soluciones óptimas, representadas por los denominados frentes de Pareto, y utilizan métricas distintas que las empleadas en los algoritmos de optimización mono-objetivo.

## 2.2 Paradigmas

Cuando para un problema no se puede asegurar que sea posible encontrar la solución óptima en un tiempo polinómico, se ha de recurrir a las **heurísticas** para su resolución.

La palabra heurística proviene del griego *heuriskein*, que significa encontrar o descubrir. Hoy en día la palabra heurística se refiere a aquella técnica o estrategia que permita mejorar las soluciones de un problema [Rayward 1996]. Existen diversos paradigmas o clasificaciones dentro de las heurísticas de resolución de problemas de optimización, entre las que destacan:

### **Búsqueda tabú.**

En la búsqueda tabú [Hajji 2004] se utiliza una memoria para guiar la búsqueda, ya que la heurística almacena las soluciones más recientes.

### **Recocido simulado.**

El recocido simulado (*Simulated Annealing*) [Sekihara 1992] es un tipo de heurística que se basa en el proceso físico de enfriamiento de un sólido, en el que se van reorganizando sus partículas hasta conseguir llegar a estados de máxima estabilidad. Son algoritmos que requieren de una temperatura inicial, final y una función de variación de la temperatura.

### **Escalando la colina.**

En las técnicas basadas en el llamado escalado de colinas, ciertas operaciones se aplican a un punto, y se generan varios estados posibles alrededor de él, de entre los cuales se escoge al mejor de ellos.

### **Computación evolutiva.**

Este término engloba a un conjunto de heurísticas que basan su funcionamiento en el mecanismo de selección natural propuesto por Charles Darwin.

Actualmente la utilización de heurísticas ha alcanzado un gran auge y difusión para abordar problemas de optimización, y entre ellas destacan principalmente los algoritmos evolutivos, que se detallan más adelante.

## 2.3 Optimización mono-objetivo y multi-objetivo

Para entender qué es la optimización mono-objetivo y multi-objetivo, en primer lugar se puede decir que las variables de decisión son aquellos valores numéricos que se eligen para un problema de optimización dado. Estos valores pueden ser denotados como:

$$x_j \text{ para } j = 1, 2, \dots, n$$

Por tanto, el vector de  $n$  variables de decisión  $\vec{x}$  es representado por:

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$

### 2.3.1 **Optimización mono-objetivo**

Los problemas de optimización mono-objetivo son aquellos que tienen un único objetivo a optimizar.

Un problema de optimización mono-objetivo se puede expresar matemáticamente de la siguiente forma: Minimizar la función objetivo  $f(x)$  sujeto a:

- $n$  restricciones de desigualdad:

$$(1) g_i(\vec{x}) \leq 0, \text{ para } i=1, \dots, n$$

- $m$  restricciones de igualdad:

$$(2) h_j(\vec{x}) = 0, \text{ para } j=1, \dots, m$$

donde  $\vec{x}$  es el vector solución.

### 2.3.2 **Optimización multi-objetivo**

A diferencia de los problemas mono-objetivos en los que la solución del problema es un punto único, en los problemas multi-objetivo no existe una solución única [Coello 2002], sino un conjunto de soluciones válidas, de las cuales ninguna es mejor que la otra. En optimización multi-objetivo se trata de encontrar un compromiso entre las distintas funciones objetivo, por lo que se suele obtener un conjunto de soluciones al que se denomina conjunto de óptimos de Pareto.

A continuación se enumera una serie de definiciones muy utilizadas en el campo de la optimización multi-objetivo:

#### 2.3.2.1 **Definición del problema**

Un problema de optimización multi-objetivo se puede definir de forma matemática como encontrar el vector  $\vec{x} = [x_1^*, x_2^*, \dots, x_n^*]$  que satisfaga:

- $p$  restricciones de desigualdad:

$$(3) g_i(\vec{x}) \leq 0, \text{ para } i=1, 2, \dots, p$$

- $q$  restricciones de igualdad:

$$(4) h_j(\vec{x}) = 0, \text{ para } j=1, 2, \dots, q$$

y optimice la función vector:

$$(5) \vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})]^T$$

donde el vector de decisión de variables es

$$(6) \vec{x} = [x_1, x_2, \dots, x_n]^T$$

El conjunto de todos los valores que satisfacen las restricciones define la región de soluciones factibles  $\Omega$ , y cualquier punto  $\vec{x} \in \Omega$  es una solución factible.

### 2.3.2.2 Dominancia de Pareto

Siendo  $u$  y  $v$  dos vectores de decisión, se dice que  $u$  domina a  $v$  si:

- $u$  no es peor que  $v$  en todos los objetivos.
- $u$  es estrictamente mejor que  $v$  en, al menos, un objetivo.

Por tanto, al comparar dos soluciones A y B, sólo pueden existir tres posibles casos:

- A domina a B.
- A es dominado por B.
- A y B no se dominan (son no dominadas entre sí).

### 2.3.2.3 Óptimo de Pareto

Un punto  $\vec{x}^* \in \Omega$  es un óptimo de Pareto si:

para cada punto  $\vec{x} \in \Omega$  y  $I = \{1, 2, \dots, k\}$ , o bien  $\forall i \in I (f_i(\vec{x}) = f_i(\vec{x}^*))$  o bien, al menos, una  $i \in I | f_i(\vec{x}) > f_i(\vec{x}^*)$ .

Esta definición dice que  $\vec{x}^*$  es un óptimo de Pareto si no existe ningún vector factible  $x$  que mejore algún criterio sin causar simultáneamente un empeoramiento en al menos otro criterio [Abbass 2002].

### 2.3.2.4 Conjunto óptimo de Pareto

El conjunto óptimo de Pareto para un problema de optimización multi-objetivo dado  $f(x)$ , se define como:

$$(7) P^* = \{x \in \Omega, / \exists x' \in \Omega, f(\vec{x}) \leq f(x)\}$$

### **2.3.2.5 Conjunto no dominado**

Entre un conjunto de soluciones  $P$ , el conjunto no dominado de soluciones  $P'$  son aquellos que no son dominados por ningún miembro del conjunto  $P$ .

### **2.3.2.6 Frente de Pareto**

Para un problema multi-objetivo dado  $f(x)$ , y un conjunto de óptimos de Pareto  $P^*$ , el frente de Pareto ( $PF^*$ ) se define como:

$$(8) PF^* = \{f(x), x \in P^*\}$$

El frente de Pareto está formado por los valores en el espacio objetivo del conjunto óptimo de Pareto. Es decir, la representación de las funciones objetivo cuyos vectores son no dominados y están en el conjunto de óptimos de Pareto  $P^*$  constituyen el Frente de Pareto  $PF^*$ .

Obtener el frente de Pareto de un problema de optimización multi-objetivo es la meta final. Dado que un frente de Pareto puede contener un gran número de puntos, una buena solución debe contener un número limitado de ellos, que se encuentren lo más cerca posible del frente de Pareto exacto, y que estén uniformemente distribuidos, para que sean de la mayor utilidad posible al experto que interpreta las soluciones.

De manera general, no existe un método eficiente para encontrar el frente de Pareto. Por este motivo, se utilizan heurísticas como los algoritmos evolutivos para producir aproximaciones del frente de Pareto de un problema.

### **2.3.3 *Problemas de optimización multi-objetivo***

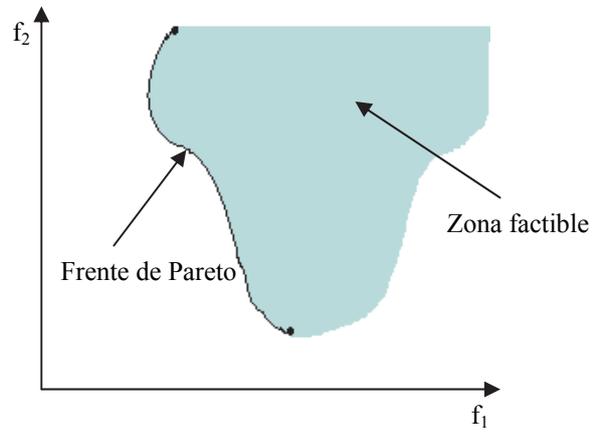
El problema de optimización multi-objetivo [Corne 2000] trata de encontrar un vector de solución que optimice simultáneamente todas las funciones objetivo, sabiendo que estas funciones se encuentran en conflicto unas con otras (son contradictorias), y mejorar una función significa empeorar otras. Algunos de estos problemas suelen convertirse a mono-objetivo (sólo un objetivo) considerando a todos los demás objetivos como restricciones del problema principal, dado que esto simplifica su solución.

La optimización multi-objetivo intenta determinar el conjunto de todos aquellos números que satisfagan las restricciones y que optimicen todas las funciones objetivo. Las restricciones son las que definen la región factible del problema y cualquier vector  $x$  que esté en esta región se considera como una solución factible.

Existen tres tipos de problemas multi-objetivo: minimizar todas las funciones objetivo, maximizar todas las funciones objetivo, o minimizar algunas y maximizar las funciones objetivo restantes.

Por simplicidad, lo más habitual es considerar únicamente problemas del primer tipo (donde se minimizan todas las funciones objetivo), transformando las funciones objetivo, en caso de ser necesario.

En resumen, el objetivo principal de la optimización multi-objetivo es descubrir soluciones tan próximas al Frente de Pareto como sea posible (Figura 3).



*Figura 3.* Frente de Pareto en un problema de optimización con dos funciones objetivo.

## 2.4 Computación Evolutiva

La Computación Evolutiva (*Evolutionary Computing*, EC) constituye un enfoque para la resolución de búsqueda de valores óptimos utilizando modelos computacionales basados en procesos evolutivos.

Las técnicas evolutivas se basan en hechos observados en la naturaleza, donde los individuos adquieren características propias mediante el mecanismo de la evolución natural. Esta evolución existe en casi todos los organismos mediante dos procesos:

- **Selección Natural:** Determina cuáles serán los organismos que se reproducirán. Si un individuo no muestra idoneidad morirá sin reproducirse.
- **Reproducción:** Garantiza la descendencia del individuo. Para ello, se realiza la mezcla y recombinación de los genes de los individuos que se han seleccionado para reproducirse (padres), los cuales en teoría son los mejores individuos para producir una nueva generación (hijos).

Las implementaciones informáticas que se rigen por estos modelos se conocen como algoritmos evolutivos [Fogel 2003] y su propósito genérico consiste en guiar una búsqueda estocástica haciendo evolucionar a un conjunto de estructuras y seleccionando de modo iterativo las más adecuadas.

De esta manera se puede decir que los algoritmos evolutivos están diseñados de forma que imitan los procesos de evolución y selección presentes en la naturaleza, mecanismos que fomentan la aparición de estructuras orgánicas complejas y bien adaptadas a su entorno.

No es extraño utilizar modelos presentes habitualmente en la naturaleza para resolver problemas en el entorno informático; como por ejemplo el Recocido Simulado (*Simulated Annealing*) [Laarhoven 1987], que se basa en el proceso físico de enfriamiento de un sólido, en el que se van reorganizando sus partículas hasta conseguir llegar a estados de máxima estabilidad, o las Redes Neuronales, en las que se trata de imitar el comportamiento de las neuronas biológicas, responsables de la conducta inteligente.

La computación evolutiva destaca por realizar un tratamiento numérico de la información. Los algoritmos evolutivos se pueden encuadrar como una subclase de las técnicas conocidas como *Soft Computing* o *Inteligencia Computacional* [Tettamanzi 2001]. Según esta clasificación, la Computación Evolutiva, las Redes de Neuronas Artificiales y la Lógica Difusa o *Fuzzy Logic*, junto con algunas otras técnicas novedosas, constituyen los campos de trabajo donde se busca la solución de problemas de optimización a través de técnicas algorítmicas basadas en la representación numérica del conocimiento.

### 2.4.1 **Algoritmos Evolutivos**

Como ya se comentó en el apartado anterior, en la naturaleza existe infinidad de organismos adaptados a la supervivencia, prácticamente en cualquier ecosistema,

o lo que es lo mismo, en casi cualquier condición. Este hecho puede resultar sorprendente si se considera que el medio ambiente está en continua evolución y sufriendo cambios de forma constante, lo que propicia la extinción de algunas especies y la adaptación de otras.

Los algoritmos evolutivos surgen para tratar de imitar este fenómeno, en el que los individuos que forman la población evolucionan sin ser conscientes de ello hacia individuos mejores. Además, los individuos son capaces de adaptarse a las nuevas condiciones que se puedan dar, debido a posibles cambios del ecosistema en el que se encuentran.

Los algoritmos evolutivos se caracterizan porque mantienen un conjunto de soluciones potenciales a un problema. Dichas soluciones son utilizadas para producir nuevas soluciones potenciales mediante la aplicación de una serie de operadores (operadores evolutivos). Dichos operadores actúan sobre algunas soluciones que han sido seleccionadas por su grado de bondad con respecto al problema tratado. Este proceso se repite cíclicamente hasta que se alcanza un cierto criterio de finalización o parada, momento en el cual se extrae el individuo que representa la solución óptima al problema.

#### **2.4.2 Conceptos generales**

Los algoritmos evolutivos son útiles para la solución de problemas NP, ya que trabajan rápidamente en grandes espacios de soluciones para operar sobre varias soluciones simultáneamente.

Para resolver un problema NP, las únicas técnicas seguras consisten en ir evaluando exhaustivamente todas las posibilidades hasta encontrar una solución óptima, pero esta forma de proceder es claramente descartable cuando el tamaño del espacio de soluciones exige unos esfuerzos computacionales inasumibles.

Los algoritmos evolutivos tienen unas características que se pueden resumir en las siguientes:

- Buscan a partir de una población de distintas soluciones.
- Utilizan una función de adaptación para realizar la búsqueda.
- Utilizan reglas de transición probabilísticas, no reglas deterministas.
- No necesitan grandes requisitos matemáticos para la solución de los problemas.
- Se trata de algoritmos evolutivos que pueden manejar cualquier tipo de funciones objetivo, y cualquier tipo de restricciones definidas en espacios de búsqueda discretos, continuos o mixtos.
- Son algoritmos de funcionamiento intrínsecamente paralelo, con todas las ventajas que ello supone, como la ejecución multiprocesador.
- Tienen mecanismos para disminuir la posibilidad de caer en óptimos locales.

- Las soluciones halladas dependen de los valores aleatorios generados al comenzar el algoritmo (población inicial).
- Introducen un cierto grado de aleatoriedad en la búsqueda de la solución.
- No siguen ningún camino de búsqueda predeterminado por el espacio de soluciones.

En la Figura 4 se muestra la estructura genética de los individuos que forman parte de una población en los algoritmos evolutivos.

A continuación, se presentan los principales conceptos que se utilizan al aplicar los algoritmos evolutivos.

- **Cromosomas:** Se trata de una cadena finita de valores definidos en un rango determinado. Contiene la información codificada de una posible solución.
- **Gen:** Es cada uno de los elementos que forman la cadena del cromosoma.
- **Alelo:** Es cada uno de los valores (uno o más) del gen.
- **Individuo o genotipo:** Cada individuo contendrá una posible solución al problema. Está formado por uno o más cromosomas.
- Valor de adecuación o **fitness** (también conocido a veces como **coste**): Valor numérico que se asigna a cada individuo para poder utilizar un criterio que evalúe la proximidad del mismo a una solución óptima.
- **Dominancia:** Es la función definida para devolver un único cromosoma a partir de todos los cromosomas que contiene el individuo. La dominancia es necesaria para poblaciones en las que los individuos están formados por dos o más cromosomas, cuya aplicación es previa para poder obtener así el valor de adecuación de los individuos.
- **Población o fenotipo:** Se trata de una expresión (decodificación) del genotipo en el dominio del problema.
- **Mutación:** Consiste en el cambio del valor de un alelo por otro valor aleatoriamente elegido.
- **Reproducción:** La reproducción consiste en elegir varios individuos para generar descendientes a partir de ellos.

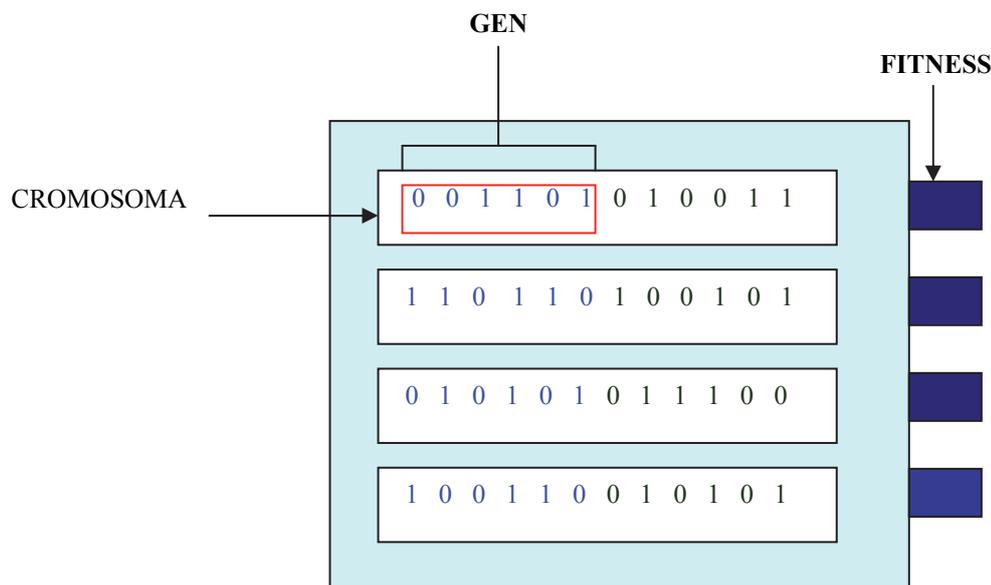


Figura 4. Estructura genética de individuos en algoritmos evolutivos.

### 2.4.3 Fundamentos de los algoritmos evolutivos

Se han mencionado anteriormente los dos procesos primarios en los que se basan los algoritmos evolutivos, y que son la selección natural y la reproducción.

El proceso de la *selección natural* se corresponde con los estudios desarrollados por Darwin en su “*Teoría de la Selección Natural*”, mientras que el proceso de la Reproducción se puede encontrar estudiado en los trabajos de Mendel sobre “*Herencia Genética*”. De ambos trabajos se puede extraer una serie de puntos:

- La evolución es un proceso que no opera sobre organismos, sino que actúa sobre sus cromosomas, que son unos instrumentos orgánicos que contienen codificada toda la información referente al individuo. Estos cromosomas y, por tanto, la información que contienen, son heredados por individuos de otra generación mediante los procesos reproductivos.
- Durante la etapa de reproducción se produce el proceso evolutivo; cuando se genera un individuo nuevo, éste hereda la información de los cromosomas de sus progenitores mediante la recombinación de genes. Asimismo, también se pueden dar en este proceso casos de mutación de genes, que propician la variabilidad del material genético de los individuos, y que pueden ser determinantes de la adaptación del individuo ante posibles cambios del entorno en el que se encuentre.
- Mediante la selección natural se permite que los mejores individuos, es decir, aquellos con información en su cromosoma más adecuada para el medio en el que se encuentran, tengan más facilidad para reproducirse que aquellos individuos que están menos adaptados al medio. De esta forma, se propicia la extinción de estos últimos individuos.

#### **2.4.4 Estructura de un algoritmo evolutivo**

Un algoritmo evolutivo es un proceso iterativo que opera sobre un conjunto  $P$  de individuos que forman una población, cada uno de los cuales tiene uno o más cromosomas almacenando toda su información.

Los cromosomas de los individuos pueden utilizarse como representación de una posible solución al problema considerado. Por tanto, cada individuo se convierte en una potencial solución al problema tratado.

Mediante un proceso adicional de codificación / decodificación ( $\rho$ ) se puede obtener la información que se codifica en los cromosomas que los individuos contienen. Esta población de cromosomas puede ser generada inicialmente de forma aleatoria o bien utilizando alguna heurística como ayuda.

El algoritmo evolutivo cuenta con una función (función de adecuación o función de fitness) para asignar a cada individuo de la población un valor de adecuación (o valor de fitness), que es un valor que depende de la información contenida en los cromosomas de cada individuo. Este valor de fitness tener una idea comparativa y cuantitativa de lo mejor o peor que es un individuo con relación a los demás. Este es el valor por el que el algoritmo evolutivo regirá su estrategia de búsqueda.

El funcionamiento general de un algoritmo evolutivo se puede explicar de la siguiente forma:

- Se parte de un conjunto de individuos o genotipos, que forman la población también denominada fenotipo.
- A partir del fenotipo se obtiene el conjunto de cromosomas de todos los individuos que lo forman.
- Sobre los genotipos que componen la población se realizan sucesivamente las operaciones de selección, reproducción, mutación y reemplazo, que se realizan de forma iterativa hasta que se cumple un criterio de terminación.

Mediante estas operaciones se promociona que los individuos más aptos de la población se puedan reproducir y, por tanto, irán transmitiendo su información con más facilidad que los menos aptos, de manera que la población evoluciona globalmente hacia individuos más aptos.

Por otro lado, el operador mutación emula a la mutación genética introduciendo un grado de aleatoriedad que evita al algoritmo caer en soluciones locales.

De esta forma, se consigue integrar en este modelo los principios de selección natural y herencia genética. Como se puede observar en la Figura 5, los pasos por los que se rige un algoritmo evolutivo durante su ejecución son:

1. Genera una población inicial de individuos de forma aleatoria.
2. Asigna un valor a cada individuo generado mediante una función objetivo para el cálculo del fitness, con el fin de determinar la proximidad o lejanía de dicha solución con respecto al óptimo.

3. Elige los individuos progenitores de acuerdo a su valor de adecuación mediante algún mecanismo de selección definido para el problema.
4. Los individuos progenitores se recombinan o se mutan para formar los individuos de la nueva generación.
5. Estos nuevos individuos se insertan en la población, bien sustituyendo la anterior población por completo o bien conviviendo con sus progenitores.
6. Estos pasos, que se corresponden con la creación de una nueva generación, se repiten hasta que se cumpla un cierto criterio de parada.

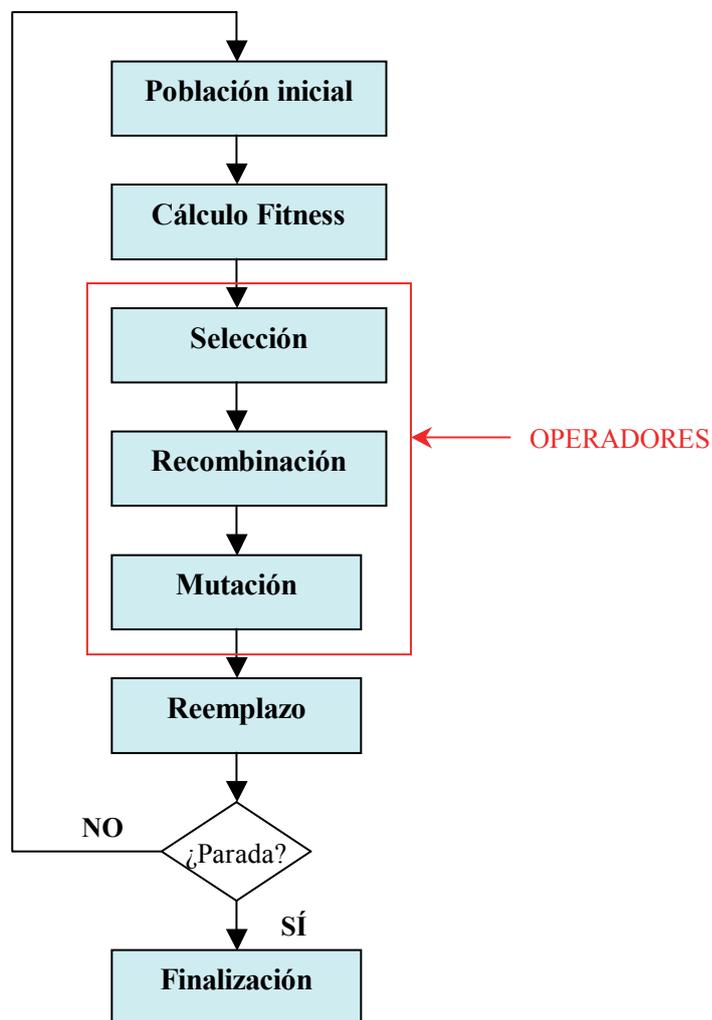


Figura 5. Fases y operadores de un algoritmo evolutivo genérico.

La Figura 6 muestra el pseudocódigo correspondiente a un algoritmo evolutivo.

```
1: t ← 0; {Contador de generaciones}
2: P[0] ← Inicializar_Población ();
3: while Not Criterio_finalización (P[t]) do
4:   Evaluar_Población(P[t]); {Evaluación de la
                             población inicial}
5:   P´[t] ← Seleccionar_Progenitores (P[t]);
6:   P´[t] ← Aplicar_Operadores_Reproducción (P´[t]);
7:   P[t+1] ← Reemplazar (P[t], P´[t]); {Población de la
                                       generación siguiente}
8:   t ← t + 1;
9: end while
10: SOLUCIÓN ← MEJOR(P[t], ∀t)
```

Figura 6. Pseudocódigo de un Algoritmo Evolutivo.

### 2.4.5 Tipos de algoritmos evolutivos

De entre las familias más importantes que han surgido del estudio de los algoritmos evolutivos se puede destacar:

- Algoritmos genéticos [Goldberg 1989].
- Estrategias evolutivas [Rechenberg 1973].
- Programación evolutiva [Fogel 2003].
- Programación genética [Koza 1992].
- Sistemas clasificadores [Holland 1967].
- Evolución diferencial [Price 2004].

Estos sistemas se diferencian principalmente en la estructura de los individuos y en el tipo de tratamiento al que se les somete.

### 2.4.6 Ventajas de los algoritmos evolutivos

Las principales ventajas que presenta la utilización de los algoritmos evolutivos en la resolución de problemas de optimización son, entre otras, las siguientes:

- Operan sobre una población (o conjunto de soluciones), lo que evita que la búsqueda se quede atascada en óptimos locales.
- No requieren conocimiento previo sobre el problema a resolver.
- Pueden combinarse con otras técnicas de búsqueda para mejorar su cometido.
- Permiten su paralelización de forma sencilla.
- Son conceptualmente fáciles de implementar y utilizar.

## 2.5 Algoritmos genéticos

Los **Algoritmos Genéticos (AGs)** son métodos estocásticos de búsqueda que han sido aplicados con éxito en muchos problemas de búsqueda, optimización y aprendizaje máquina [Goldberg 1989][Reeves 2003].

A diferencia de otras técnicas de optimización, los AG mantienen una población de soluciones codificadas (individuos) que son manipuladas competitivamente mediante operadores de variación para encontrar un óptimo global. El AG funciona mejor que otros algoritmos de optimización cuando la función objetivo es discontinua, no diferenciable, estocástica o altamente no lineal.

Podemos decir que el AG difiere de un algoritmo de optimización estándar en dos características principalmente, tal como se describe en la Tabla 2.

Algoritmo de optimización estándar	Algoritmo genético
* Genera un único punto en cada iteración. * La secuencia de puntos se aproxima a una solución óptima.	* Genera una población de puntos en cada iteración. * La población se aproxima a una solución óptima.
* Selecciona el próximo punto en la secuencia mediante computación determinista.	* Selecciona la próxima población mediante cálculos que implican selecciones aleatorias.

Tabla 2. Diferencias entre algoritmo estándar y algoritmo genético.

### 2.5.1 Codificación de las soluciones y función de fitness

En los AG los individuos son posibles soluciones del problema de optimización, que están codificadas en forma de cromosoma. Un cromosoma es una estructura de datos formada por genes, que corresponden a parámetros del problema. Se puede definir al individuo como cualquier punto para el que se aplicarla función de *fitness*. El valor de la función de *fitness* para un individuo es su puntuación.

La función de *fitness* es la función que puntúa al individuo de la población, y generalmente se quiere minimizar o maximizar su valor. Para los algoritmos de optimización estándar, esta función es conocida también como función objetivo.

### 2.5.2 Población, generaciones y diversidad

Una población es un *array* de individuos. El mismo individuo puede aparecer más de una vez en la población. En cada iteración, el AG realiza una serie de computaciones en la población actual para producir una nueva población, en el transcurso de lo que se denomina generación.

El término *diversidad* se refiere a la distancia (grado de diferenciación) media entre los individuos en una población. Una población tiene alta diversidad si la distancia media es grande; en caso contrario, la población tiene baja diversidad. La diversidad es esencial en el AG porque le permite buscar la solución óptima en una región más grande del posible espacio de búsqueda.

### 2.5.3 Operadores de un algoritmo genético

El AG es un método para resolver problemas de optimización basado en la selección natural, que es el proceso maestro de la evolución biológica. El AG modifica de forma cíclica en el tiempo (a lo largo de generaciones) una población de soluciones, donde se seleccionan individuos para ser padres y ser utilizados para producir hijos para la próxima generación. Sobre las generaciones sucesivas, la población “evoluciona” hacia una en la que estaría la solución óptima.

El AG utiliza básicamente tres operadores, llamados operadores genéticos, en cada generación para crear la próxima población:

- **Selección:** selecciona los individuos, llamados padres, que contribuyen a generar la descendencia para crear la población de la próxima generación.
- **Cruce** (*crossover*): combina dos padres para formar la descendencia.
- **Mutación:** aplica aleatoriamente cambios a los individuos (generalmente, a los padres) para introducir un cierto grado de diversidad o incertidumbre.

### 2.5.4 Funcionamiento de un algoritmo genético

Básicamente, el AG tiene la siguiente forma de proceder:

- El algoritmo comienza creando una población inicial aleatoriamente.
- El algoritmo entra en su primera generación aplicando los tres operadores genéticos para crear una nueva población, la cual será la población inicial para la siguiente generación. Un esquema usual para aplicar los operadores genéticos consiste en los siguientes pasos:
  - Puntuar cada individuo de la población actual mediante la llamada función de *fitness* o función de coste.
  - Seleccionar los padres según sus valores de *fitness*.
  - Generar la descendencia (hijos, nuevos individuos) a partir de los padres seleccionados. Los hijos son producidos por una de estas dos formas:
    - Haciendo cambios aleatorios a un único padre (mutación).
    - Combinando o intercambiando los genes de dos padres (cruce).
  - Reemplazar la población inicial de la generación con los hijos para formar la población utilizada en la próxima generación.

Un AG tiene unos parámetros que han de fijarse en cada ejecución, destacando:

- **Tamaño de la población.** La población debe ser lo suficientemente grande como para garantizar la diversidad de las soluciones.
- **Condición de terminación o parada.** Es la condición que, de cumplirse, determina cuándo el AG deja de evolucionar. Suelen ser la convergencia del AG, un número determinado de generaciones o el tiempo transcurrido.

## 2.6 Evolución Diferencial

La **Evolución Diferencial** (*Differential Evolution*, **DE**) es una heurística evolutiva desarrollada por Rainer Storn y Kenneth Price [Storn 1995] para optimización en espacios continuos. Esta heurística nació con los intentos de Kenneth Price por resolver el polinomio de Chebychev, problema propuesto por Rainer Storn. En estos intentos fue cuando Kenneth Price concibió la idea de utilizar las diferencias entre los vectores para perturbar el vector de la población.

Con esta idea principal, Storn y Price comenzaron a realizar simulaciones y pruebas para poder construir un método versátil y robusto como lo es hoy en día esta heurística. La comunidad de Evolución Diferencial ha estado creciendo desde 1995 y cada día más investigadores trabajan con esta técnica [Price 2004].

En la evolución diferencial, las variables se representan mediante números reales. La población inicial se genera de forma aleatoria y se seleccionan tres individuos como padres. Uno de ellos es considerado como el padre principal, el cual se perturba con el vector de los otros dos. Si el valor resultante es mejor que el elegido para ser sustituido, entonces el padre (que se considera principal) se reemplaza. En caso contrario, se retiene al padre principal.

### 2.6.1 Esquema de Evolución Diferencial

La evolución diferencial es un método de búsqueda en el cual la población utiliza  $N$  vectores en cada generación  $G$ :  $x_{i,G}$ , para  $i = 0, 1, \dots, N-1$ , donde el valor de  $N$  no cambia durante el proceso de minimización.

La población inicial se elige de manera aleatoria.

La idea principal es un nuevo esquema para generar vectores. La Evolución Diferencial genera estos nuevos vectores cuando se suma la diferencia de pesos entre dos vectores miembros de la población a un tercer vector miembro.

Si la aptitud (fitness) del vector resultante es menor que el miembro de la población elegido, entonces el nuevo vector reemplaza al vector con el cual fue comparado. El mejor vector  $x_{mejor,G}$  se evalúa en cada generación  $G$ .

#### 2.6.1.1 Mutación

Para cada vector  $\overrightarrow{x_{l,G}}$ , un vector perturbado  $\overrightarrow{v_{l,G+1}}$  se genera mediante (9).

$$(9) \overrightarrow{v_{l,G+1}} = \overrightarrow{x_{r1,G}} + F * (\overrightarrow{x_{r2,G}} - \overrightarrow{x_{r3,G}})$$

donde los índices  $r1, r2, r3 \in [0, 1, \dots, N-1]$  son diferentes entre sí y también diferentes respecto al índice  $i$  del vector actual ( $r1 \neq r2 \neq r3 \neq i$ ).

Los enteros  $r1, r2$  y  $r3$  son seleccionados aleatoriamente en el intervalo  $[0, N-1]$ .

El factor de ponderación (*weighting factor*)  $F$  es un valor real y constante que controla la amplificación en la variación diferencial  $\overrightarrow{x_{r2,G}} - \overrightarrow{x_{r3,G}}$ .

$F \in [0,1+]$  es un parámetro de control en evolución diferencial.

### 2.6.1.2 Cruce

Los siguientes vectores forman parte de la fase de cruce (recombinación): El vector perturbado  $\vec{v}_{i,G+1} = (v_{1,i,G+1}, v_{2,i,G+1}, \dots, v_{D,i,G+1})$  y el vector de la población  $\vec{x}_{i,G} = (x_{1,i,G+1}, x_{2,i,G+1}, \dots, x_{D,i,G+1})$  intervienen en la fase de cruce.

El operador cruce se utiliza para incrementar la diversidad de los resultados en los vectores. Con este propósito, se utiliza el vector prueba  $\vec{u}_{i,G+1} = (u_{1,i,G+1}, u_{2,i,G+1}, \dots, u_{D,i,G+1})$ , con:

$$(10) \quad \vec{u}_{j,i,G+1} = \begin{cases} v_{j,i,G+1}, & \text{si } (\text{rand}_j[0,1] < CR) \text{ ó } (j = k) \\ (x_{j,i,G})_j, & \text{en cualquier otro caso} \end{cases}$$

donde  $i = 0, 1, \dots, N-1$ ,  $j = 0, 1, \dots, D-1$ , siendo  $D$  el número de parámetros.  $CR \in [0, 1]$  es la probabilidad de cruce y constituye una variable de control para el esquema DE. El índice  $k \in \{1, \dots, n\}$  (elegido de forma aleatoria) asegura que el vector  $\vec{u}_{i,G+1}$  diferirá de su padre  $\vec{x}_{i,G}$  en, al menos, un parámetro.

La ecuación (10) significa que una cierta secuencia de elementos del vector en  $\vec{u}_{i,G+1}$  son elementos idénticos a  $\vec{v}_{i,G+1}$ , y los demás elementos de  $\vec{u}_{i,G+1}$  adquieren el valor original de  $\vec{x}_{i,G}$ .

En la Figura 7 se muestra un ejemplo para  $D = 5$ . El valor aleatorio  $k$  se recalcula para cada nuevo vector de prueba.

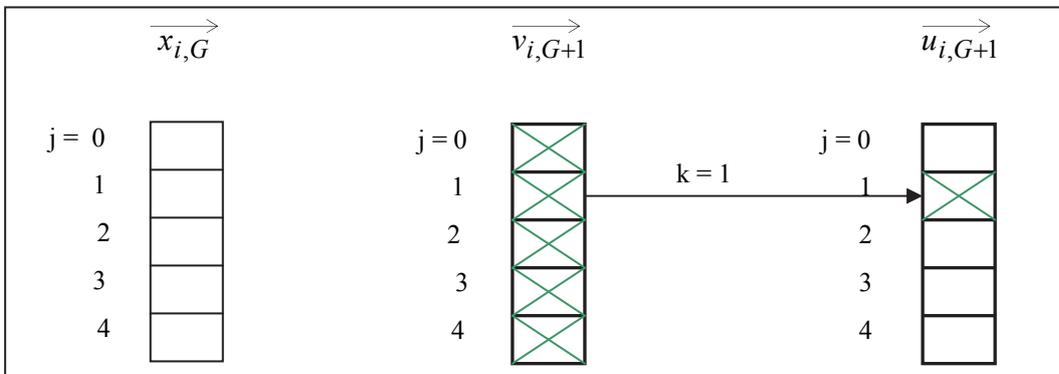


Figura 7. Representación del proceso de cruce para  $D = 5$ .

### 2.6.1.3 Selección

Para decidir si el nuevo vector  $\vec{u}_{i,G+1}$  se convierte en miembro de la población de la generación  $G+1$ , se compara con el vector  $\vec{x}_{i,G}$ , y si tiene un menor valor al evaluarlo en la función objetivo, entonces  $\vec{u}_{i,G+1}$  se hace  $\vec{x}_{i,G+1}$ , de otra forma el valor de  $\vec{x}_{i,G}$  es retenido, según se expresa en la ecuación (11).

$$(11) \quad \vec{x}_{i,G+1} = \begin{cases} u_{i,G+1}, & \text{si } (f(u_{i,G+1}) < f(x_{i,G})) \\ (x_{i,G})_j, & \text{en cualquier otro caso} \end{cases}$$

Las distintas fases de la Evolución Diferencial para generar un nuevo vector en la población, y determinar si pasa a la siguiente generación se ilustran en la Figura 8:

1. Se elige un vector de forma aleatoria dentro de la población. Supóngase que dicho vector tiene un valor de fitness de 94.
2. Se seleccionan dos vectores aleatoriamente. Supóngase que uno de ellos tiene un fitness de 5 y otro de 62.
3. Se calcula la diferencia en sus pesos, dando origen a un nuevo vector con un valor de fitness diferente.
4. Un tercer vector es escogido de forma aleatoria de la población. En este caso, tiene una aptitud de 77 y se realiza una suma de éstos.
5. Este vector de prueba obtenido en el paso cuatro se cruza con el vector original, para obtener así un nuevo individuo que será el hijo resultante del cruce. Este nuevo vector posee un fitness de 44.
6. Por último, se compara el vector padre y el vector hijo para determinar cuál de los dos continúa formando parte de la próxima generación. En este ejemplo, el nuevo vector generado se copia en la nueva generación porque el vector\_padre (94) es mayor que el vector\_hijo (44)

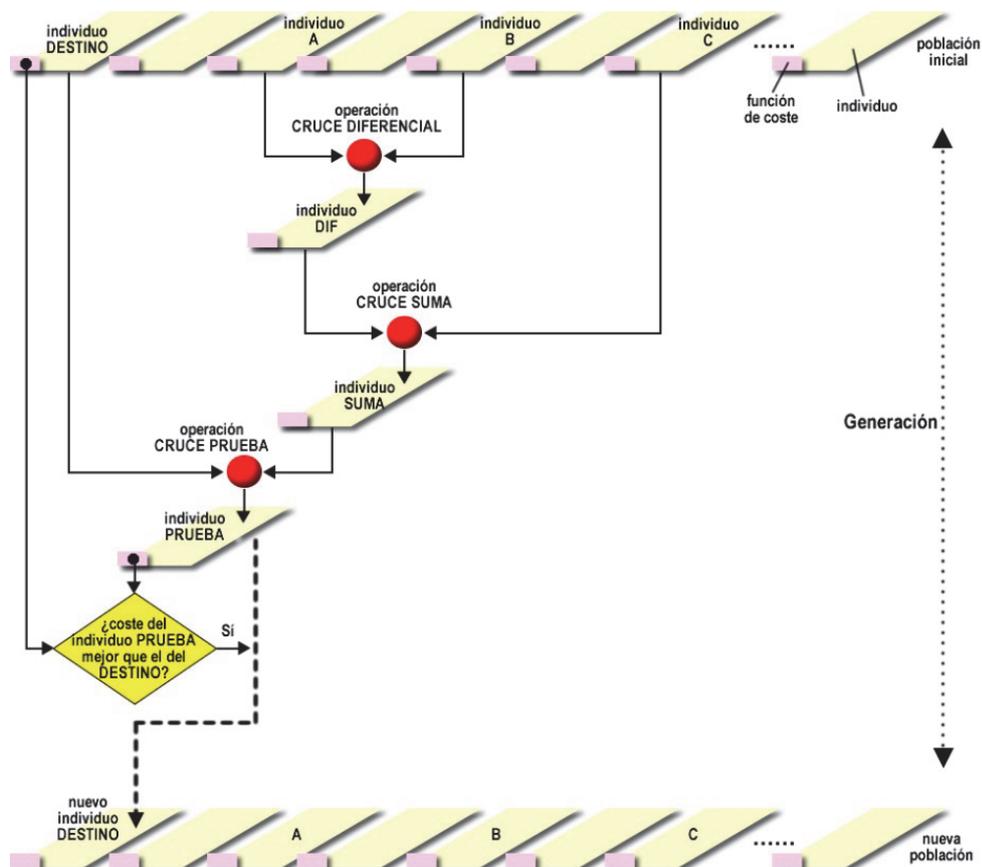


Figura 8. Ejemplo bidimensional de una función objetivo mostrando el proceso de generación en el esquema de Evolución Diferencial.

## 2.6.2 Estrategias de Evolución Diferencial

Las diferentes estrategias que pueden adoptarse en el algoritmo de Evolución Diferencial dependen del tipo de problema al que se aplique. Las estrategias se basan en el vector a perturbar, el número de diferentes vectores considerados para la perturbación, y finalmente el tipo de cruce a utilizar.

A continuación, se listan las diez estrategias propuestas por Price & Storn:

1. *ED/mejor/1/exp*
2. *ED/aleatorio/1/exp*
3. *ED/aleatorio-al-mejor/1/exp*
4. *ED/mejor/2/exp*
5. *ED/aleatorio/2/exp*
6. *ED/mejor/1/bin*
7. *ED/aleatorio/1/bin*
8. *ED/aleatorio-al-mejor/1/bin*
9. *ED/mejor/2/bin*
10. *ED/aleatorio/2/bin*

La convención utilizada anteriormente es: ED/x/y/z:

- **ED** se refiere a la Evolución Diferencial.
- **x** representa el vector a perturbar; puede ser el mejor vector de la generación anterior (mejor) o cualquier vector elegido aleatoriamente (aleatorio).
- **y** es el número de vectores considerados para perturbar *x*. Para la perturbación de un simple vector, se eligen aleatoriamente 3 diferentes vectores, y la diferencia de dos de ellos se suma al tercero. Para la perturbación de dos vectores diferentes, se eligen 5 vectores distintos; la diferencia de pesos para cada par de cualquiera de los 4 primeros vectores se suma al quinto.
- **z** se entiende como el tipo de cruce utilizado (*exp*: exponencial; *bin*: binomial). El cruce en DE se realiza en *d* variables del vector dentro de un ciclo limitado por el valor de *CR*. En el cruce exponencial, la primera vez en la que un número aleatorio entre (0,1) supera el valor de *CR* se suspende este ciclo y las variables que queden por alterar quedan intactas. En el cruce binomial, el cruce se realiza en cada una de las *d* variables si al elegir un número aleatorio entre (0,1) éste sea menor que el valor de *CR*. Así, para valores altos de *CR* el cruce exponencial y la binomial se comportan de forma similar.

La estrategia a adoptar para cada problema se determina de forma independiente mediante ensayo. De esta forma, una estrategia que funcione mejor para resolver un problema puede no trabajar bien cuando se aplica para otro problema distinto.

Así también, los parámetros que se adoptan para un problema tienen que ser determinados mediante ensayo. Sin embargo, la estrategia (*ED/aleatorio/1/bin*) es la más utilizada, sobre todo en algunos trabajos que se detallan en esta tesis.

## **2.7 Estado del arte en optimización evolutiva multi-objetivo**

Goldberg propuso utilizar la dominancia de Pareto como un criterio de selección para diseñar algoritmos evolutivos multi-objetivo. La idea principal es marcar a las soluciones que son no dominadas, asignarles una jerarquía más alta y retirarlas de la población. Este proceso continúa con las siguientes soluciones que son no dominadas y se les asigna otra jerarquía menos alta que la primera.

Este proceso se realiza hasta que se ha asignado una jerarquía a cada individuo de la población [Dontas 1990]. También se sugirió el uso de nichos para evitar la convergencia a un solo punto.

A continuación se muestran los algoritmos más representativos que utilizan selección basada en una jerarquización de Pareto.

### **2.7.1 MOGA**

*Multi-Objective Genetic Algorithm (MOGA)* fue propuesto por Fonseca y Fleming [Fonseca 1993]. Según este enfoque, la jerarquía de un individuo depende del número de individuos en la población actual que lo dominan (los individuos no dominados tendrán una jerarquía de '1'). MOGA utiliza nichos para distribuir uniformemente las soluciones, y aplica restricciones en el cruce y otros mecanismos para evitar una convergencia prematura.

### **2.7.2 NSGA**

*Non-Dominated Sorting Genetic Algorithm (NSGA)* fue planteado por Srinivas y Deb [Srinivas 1994] y utiliza la idea original de Goldberg al jerarquizar la población por capas. De esta forma, los individuos que están en la primera capa tienen el máximo valor de la función de fitness (por ser no dominados con respecto a toda la población) y se obtienen muchas copias de estos individuos con respecto al resto de la población.

El algoritmo NSGA usa compartición de aptitud para distribuir los individuos a lo largo del frente de Pareto.

### **2.7.3 NSGA-II**

*Non-Dominated Sorting Genetic Algorithm - II (NSGA - II)* fue propuesto también por Deb et al. [Deb 2000]. Se trata de una nueva versión del NSGA, con un comportamiento más eficiente que el anterior. Añade elitismo y un operador que ayuda a mantener la diversidad sin requerir de parámetros adicionales.

#### **2.7.4 PAES**

*Pareto Achieved Evolution Strategy (PAES)* fue desarrollado por Knowles y Corne [Corne 2000]. Este algoritmo sigue una estrategia evolutiva de dos miembros, en la que un padre genera un solo hijo en combinación con un archivo externo. Este archivo almacena los individuos no dominados generados a lo largo del proceso evolutivo. Cuando se encuentra un individuo no dominado, se compara con los del archivo externo, y en caso de que sea no dominado con respecto a éstos, una malla adaptativa selecciona los individuos que se eliminan del archivo externo.

La malla ayuda a mantener diversidad y a distribuir uniformemente las soluciones no dominadas producidas. Además, no necesita parámetros y su complejidad computacional es menor que la de los nichos.

#### **2.7.5 SPEA**

*Strength Pareto Evolutionary Algorithm (SPEA)* fue propuesto por Zitzler y Thiele [Zitzler 1998]. Utiliza un archivo externo que contiene las soluciones no dominadas del problema. En cada generación, los individuos no dominados se copian al archivo externo y se les asigna un valor extra, el cual es proporcional al número de individuos que lo dominan.

Por tanto, el fitness de cada individuo se calcula de acuerdo con este valor extra de todos los individuos del archivo externo que lo dominan. Además, se adopta una técnica de cúmulos para mantener diversidad.

#### **2.7.6 SPEA-2**

El *Strength Pareto Evolutionary Algorithm 2 (SPEA 2)* fue presentado también por Zitzler et al. [Zitzler 2002]. Este algoritmo tiene tres diferencias principales con respecto a su predecesor:

- Para calcular el fitness de los individuos se tienen en cuenta la cantidad de individuos que lo dominan y la de individuos que son dominados por él.
- Utiliza una densidad vecinal que guía la búsqueda más eficientemente.
- Sigue un esquema del truncamiento del archivo externo para garantizar la preservación de las soluciones en los extremos.

# ***3 Optimización de Series Temporales***

*“El que sube una escalera  
debe empezar por el  
primer peldaño.”*

**Sir Walter Scott.**  
Novelista, poeta y abogado  
escocés (1771-1832).

### 3.1 Resumen

En este trabajo se presenta un estudio sobre la aplicación de los AGs para optimizar un parámetro que aumenta la precisión de la modelación e identificación de las **Series Temporales (TS, de *Time Series*)**, muy utilizadas en el campo del **tratamiento de la señal** y en las **comunicaciones multimedia**. El interés de aumentar la precisión de las ecuaciones matemáticas que modelan las series temporales estriba en mejorar la predicción de su comportamiento futuro.

Con este propósito, esta tesis propone un modelo de **Algoritmo Genético Paralelo (AGP)** para mejorar la precisión de los resultados. Este AGP puede encuadrarse como AG paralelo de grano grueso y es **simulado en un entorno monoprocesador**. Los resultados se han comparado con de un AG simple, teniendo en cuenta que ambos tengan las mismas características y calculando cuánto mejora el resultado AGP respecto al AG simple. Para ello se han utilizado 12 *benchmarks* de series temporales de distintos tamaños.

Para desarrollar este trabajo se han empleado técnicas de **Identificación de Sistemas (IS)** y además, se usa el **Algoritmo Recursivo de Mínimos Cuadrados (RLS)** y el modelo **Autoregressive Moving Average Extensive (ARMAX)**.

La precisión de los modelos obtenidos a partir de la IS depende del ajuste del valor de un parámetro existente en el algoritmo RLS: la constante ***forgetting-factor* (*ff* ó  $\lambda$ )**, también conocida como factor de olvido.

El problema de la optimización de  $\lambda$  en la modelación de series temporales es adecuado para ser tratado por algoritmos genéticos, pues existe un único valor óptimo de  $\lambda$  de entre las muchas posibilidades dependiendo de la precisión, ya que hay muchas series temporales cuya identificación es muy sensible a este valor. La elección del parámetro *ff* frente a otros (como el tamaño del modelo matemático) se ha basado experimentaciones previas.

La optimización del parámetro  $\lambda$  es importante porque influye en la precisión de la identificación y, a partir de ella, en una mejor predicción del comportamiento futuro de la TS. Esta precisión se justifica al abordar problemas donde se exige minimizar los errores en la estimación (sistemas críticos), o cuando tienen una gran variabilidad en su comportamiento dinámico (identificación en tiempo real).

La función de coste o *fitness* (función objetivo) que se pretende minimizar depende del valor del factor de olvido *ff*, y representa el error cometido en la estimación global de la serie temporal.

En este trabajo se ha utilizado la librería **Genetic Algorithm and Direct Search Toolbox (GADST)**, proporcionada por el entorno de desarrollo del lenguaje MATLAB [Matlab 2012], para la implementación de los algoritmos genéticos. Esta librería es una colección de funciones en lenguaje MATLAB que incluyen rutinas para resolver problemas de optimización usando AGs y búsquedas directas. Para computar los algoritmos se ha utilizado un computador que, en el momento de estos trabajos (año 2004) era de los más rápidos.

### 3.2 Introducción

Un aspecto que ha despertado gran interés en diversos campos de la ciencia y la ingeniería es encontrar modelos matemáticos que describan fiablemente el comportamiento de un sistema complejo cuya descripción matemática no es posible por otras técnicas. De estos sistemas sólo suelen conocerse sus valores de E/S, y poco de la estructura física de los mismos. Surge así la necesidad de plantear las técnicas de identificación para estos sistemas [Soderstrom 1989].

Los valores de las señales de entrada y salida (E/S) de un sistema están relacionados por la naturaleza del mismo. Para obtener un modelo matemático hay dos caminos, que también pueden combinarse:

- Por un lado, analizar las leyes físicas que gobiernan al sistema, y deducir las ecuaciones que lo describen. Esto es a menudo muy difícil, debido a la complejidad de muchos sistemas o al conocimiento incierto de la naturaleza que los constituyen.
- Otro camino utilizado es el método experimental en el que, a base de observar (medir) las entradas y salidas, pueda describirse matemáticamente el comportamiento del sistema.

El fundamento de la IS nace del método experimental, y trata de encontrar un modelo matemático paramétrico del sistema, del cual se conoce su comportamiento para unos estímulos dados. Esta tesis considera sistemas discretos en el tiempo (sistemas muestreados, con valores de E/S conocidos según un determinado periodo de muestreo) y de naturaleza SISO (*Single Input, Single Output*, una señal de entrada y una de salida).

En Figura 9 se muestra un sistema discreto SISO determinístico, es decir, discretizado en el tiempo, del cual se toman  $NM$  medidas de su salida  $y(k)$  para una entrada conocida (o medida)  $u(k)$ , y considerando una señal de ruido o perturbación  $e(k)$ . Los valores de estas señales son tomados en los instantes de muestreo  $k = 0, 1, 2, \dots, NM-1$ , donde el periodo de muestreo es  $T$ .

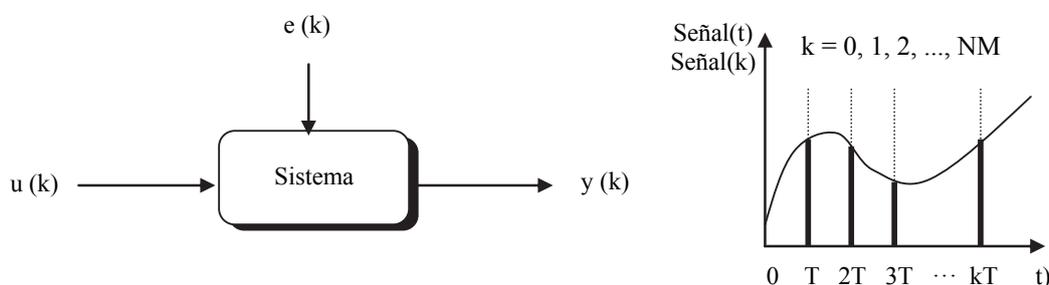


Figura 9. Sistema SISO muestreado en tiempo discreto.

En este trabajo de tesis se considera únicamente un tipo simplificado de sistema SISO: la serie temporal. La serie temporal es una señal única discreta en el tiempo sin entrada  $u(t)$ . Se simplifica también el problema no considerando el ruido.

### 3.3 Identificación y predicción de series temporales

Muchos fenómenos físicos son observados, medidos y descritos mediante una única señal  $y(k)$ , especialmente en los ámbitos del tratamiento digital de la señal y de las comunicaciones multimedia. La complejidad de muchos de estos sistemas hace imposible su modelado matemático a partir de formulaciones de sus características y comportamientos. Cuando no es posible cuantificar las ecuaciones matemáticas que describen un sistema determinado por una serie temporal, se puede abordar el problema mediante la identificación de sistemas, la cual permite un modelado basado en el comportamiento pasado de los valores observados de la señal.

La metodología de identificación que se utiliza en este trabajo se basa en una modelación matemática de la TS mediante el modelo ARMAX (descripción polinómica de grado ajustable) cuyos parámetros son obtenidos con el algoritmo recursivo de mínimos cuadrados RLS.

En la Figura 10 puede observarse un sistema cuya señal de salida  $y(k)$  representa la serie temporal que lo define. Los valores observados son procesados por un algoritmo de identificación que determina el valor de los parámetros del **modelo**  $\theta(k)$ , el cual describe al sistema (lo modela), permitiendo calcular el **valor estimado**  $ye(k)$  a partir del mismo.

A su vez, este modelo puede ser utilizado para hacer una **predicción o simulación** del comportamiento futuro del sistema, descrito por la señal  $ys$ .

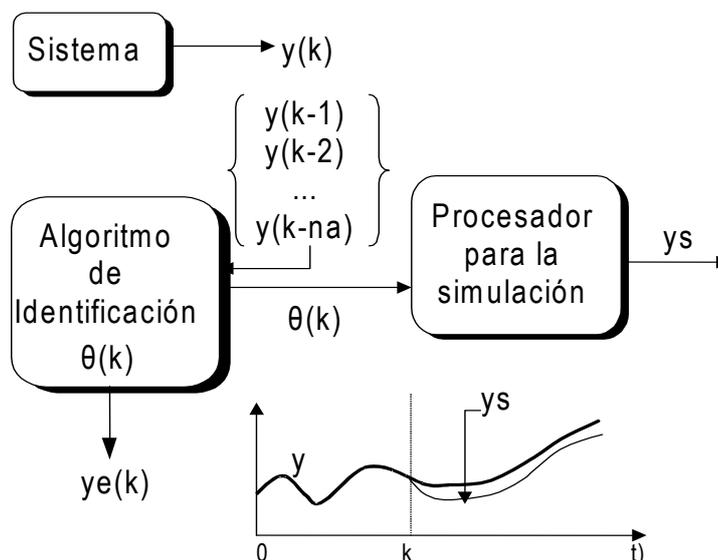


Figura 10. Tratamiento de la serie temporal. La señal muestreada  $y(k)$  es utilizada por un algoritmo de identificación que proporcionará la predicción del comportamiento futuro del sistema.

La **precisión** de los modelos de identificación depende tanto del tamaño del modelo (dimensión del polinomio que lo define) como del factor de olvido.

La identificación de sistemas, como método de predicción a muy corto plazo (a una muestra o instante  $k$ ) es muy eficaz. Con el modelo obtenido en este instante  $k$  se pueden hacer predicciones futuras, si bien cuanto mayor sea el plazo de la predicción, peor será ésta.

En la Figura 11 puede observarse un ejemplo de análisis de una serie temporal. La serie tiene un comportamiento dinámico de gran variabilidad que exige una identificación precisa (Figura 11a). Se puede constatar que la predicción de la serie temporal en los instantes futuros a partir del modelo obtenido en un instante determinado es peor cuanto mayor es el plazo de la predicción (Figura 11b). No obstante, si se limita la predicción a un instante futuro (es decir, si se re-elabora el modelo matemático de la serie temporal en cada instante), la estimación es mejor (Figura 11c), si bien perdemos información del posible comportamiento o tendencia en el futuro de la serie temporal.

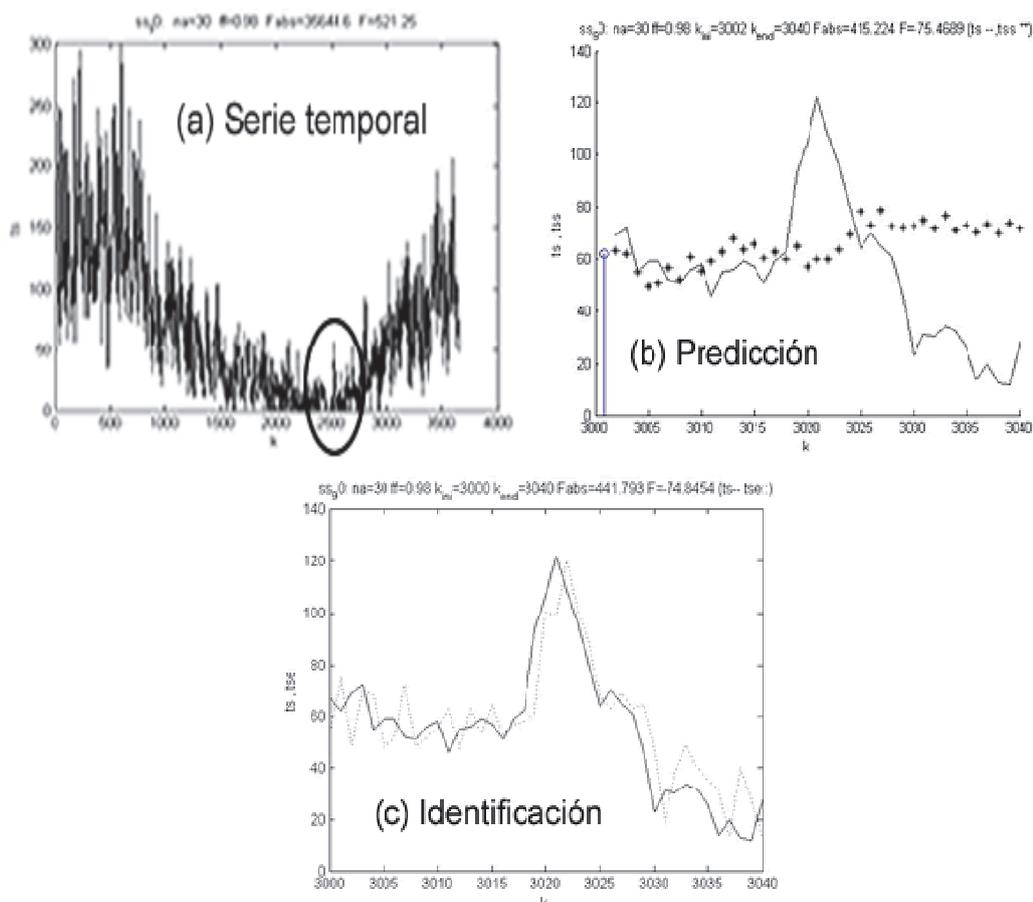


Figura 11. En (a) se muestra una TS, un detalle de la cual es sometida a predicción a largo plazo (b) y a muy corto plazo o a una muestra vista (c), que es realmente lo que supone la IS.

### 3.3.1 Modelo ARMAX

Para la modelación de las series temporales se utiliza la descripción paramétrica ARMAX [Ljung 1999] para la señal  $y(k)$ . Su definición se muestra en la ecuación (12), donde  $na$  es la dimensión del modelo ( $a_i$  son constantes).

La estimación  $y_e(k)$  de la serie en un instante  $k$  puede ser formulada como en la ecuación (13), donde  $\theta$  es un vector de constantes (ecuación (14)) y  $\varphi$  es un vector que contiene valores observados o medidos anteriormente del sistema (ecuación (15)). A partir de aquí se puede determinar el error cometido en la estimación, según se describe en la ecuación (16).

$$(12) \quad y(k) + a_1 \cdot y(k_1) + \dots + a_{na} \cdot y(k_{na}) = 0$$

$$(13) \quad y_e(k) = [-a_1 \cdot y(k-1) - \dots - a_{na} \cdot y(k-na)] = \varphi^T(k) \cdot \theta$$

$$(14) \quad \theta = \begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_{na} \end{pmatrix}$$

$$(15) \quad \varphi(k) = \begin{pmatrix} -y(k-1) \\ -y(k-2) \\ \dots \\ -y(k-na) \end{pmatrix}$$

$$(16) \quad error(k) = y(k) - y_e(k)$$

### 3.3.2 Algoritmo RLS

Como se acaba de decir, la metodología de identificación considerada está basada en el modelo ARMAX, cuyos parámetros son obtenidos a partir del algoritmo recursivo de mínimos cuadrados RLS [Ljung 1991][Parkum 1992].

El algoritmo RLS parte de unas condiciones iniciales 0, siendo  $k$  el instante actual,  $p$  una constante que denota un instante de tiempo discreto inicial,  $\theta$  la matriz de parámetros del modelo vacía, y  $P$  una matriz intermedia, siendo  $I$  la matriz identidad. A continuación se calcula la señal estimada  $y_e$  para el instante actual según el modelo ARMAX (18) y el error cometido en la estimación (19).

Tras estos pasos consecutivos, se calculan las matrices  $K$  y  $P(k)$  según las ecuaciones (20) y (21) respectivamente. Con estos valores, el algoritmo vuelve a iterar a partir de la ecuación (18) para calcular el valor estimado de la señal para el nuevo valor del instante muestreado, y así sucesivamente.

$$(17) \quad k = p, \theta(p) = 0, P(p) = 1000 I(p \geq na)$$

$$(18) \quad y_e(k) = \varphi^T(k) \cdot \theta(k-1)$$

$$(19) \quad err(k) = (y(k) - y_e(k))$$

$$(20) \quad K = \frac{P(k-1) \cdot \varphi(k)}{\lambda + \varphi^T(k) \cdot P(k-1) \cdot \varphi(k)}$$

$$(21) \quad P(k) = \frac{P(k-1) - k \cdot \varphi^T(k) \cdot P(k-1)}{\lambda}$$

Por tanto, el proceso de identificación y estimación va cambiando el modelo conforme se avanza en el tiempo discreto, haciéndolo más preciso, ya que se tiene en cuenta un número cada vez mayor de observaciones pasadas, recogiendo por tanto una mayor porción del comportamiento del sistema (Figura 12). Esto permite realizar estimaciones cada vez más precisas.

En la Figura 13 se puede observar cómo la estimación, a lo largo del tiempo, puede variar en cuanto a su precisión, variando el error cometido respecto de la medida real. Por tanto, el objetivo del tratamiento de la IS sería minimizar el error cometido en la estimación, no en un instante determinado, sino a lo largo de todo el tiempo transcurrido en la observación de la serie temporal.

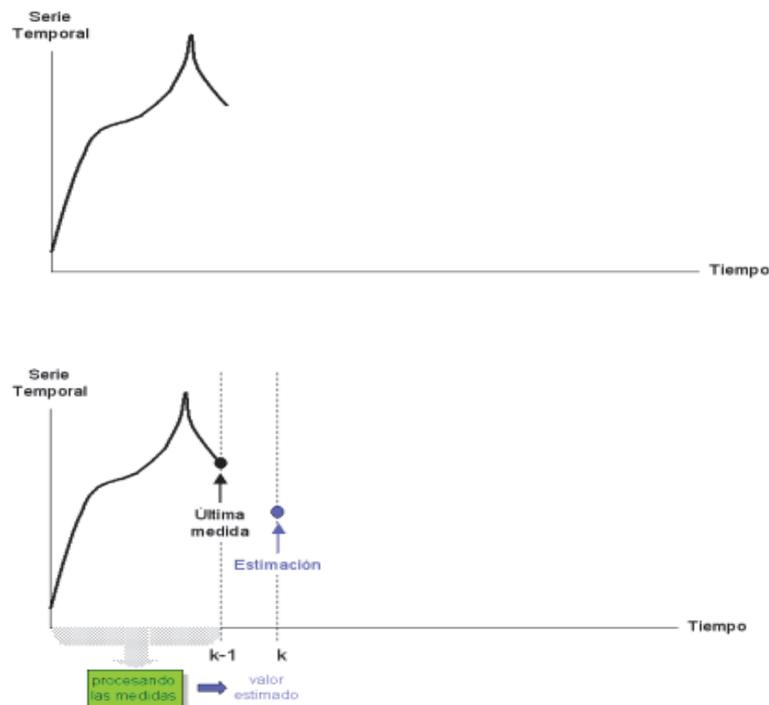


Figura 12. Identificación-estimación ARMAX - RLS. La estimación actual  $k$  tiene en cuenta las medidas observadas en instantes anteriores: cuantas más se tengan en cuenta, más preciso será el modelo. (continúa)

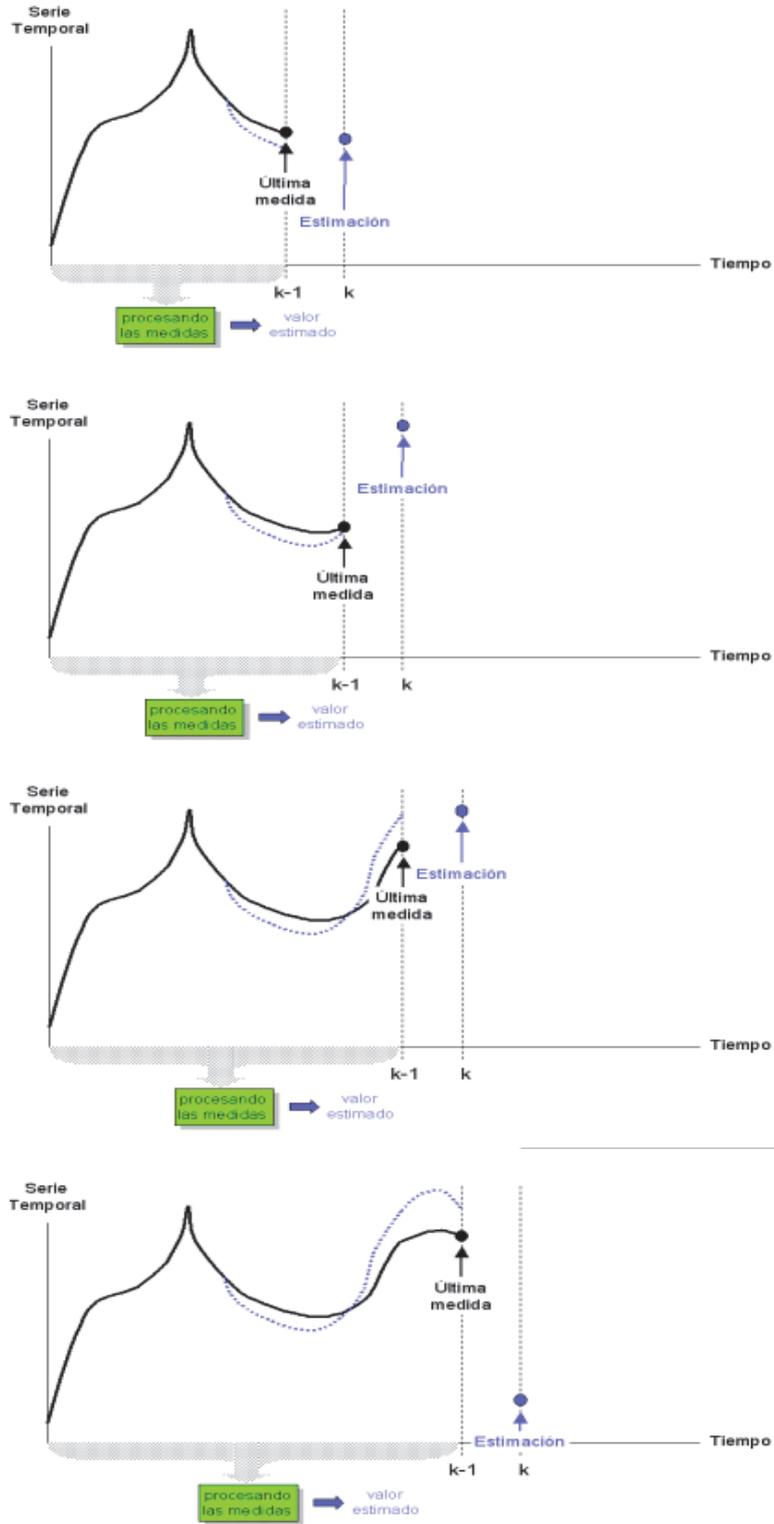


Figura 12. (continuación)

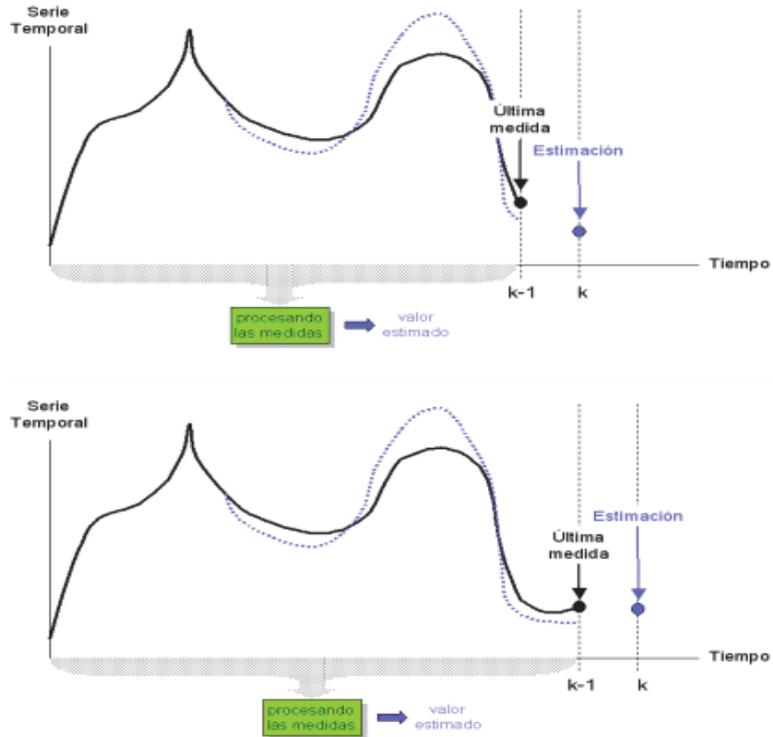


Figura 12. (continuación)

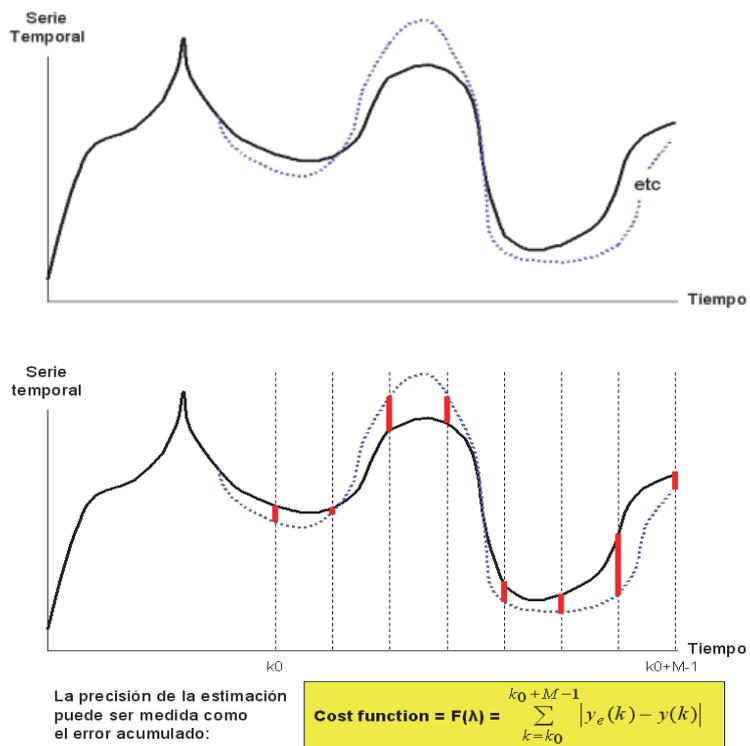


Figura 13. El error global en la estimación se basa en los errores anteriores, y es la función *fitness* que hay que minimizar para una IS más precisa.

### 3.4 Definición del problema de optimización

La precisión de los modelos obtenidos a partir de la IS depende del ajuste del valor de un parámetro existente en el algoritmo RLS: la constante *forgetting-factor*,  $\lambda$ , también conocida como factor de olvido.

El valor de este parámetro suele encontrarse en la literatura científica situado en un rango entre 0,95 y 0,98 [Ljung 1999]. No obstante, en los trabajos de esta tesis se ha constatado que cada serie temporal puede tener un valor óptimo de  $\lambda$  distinto; es más, el valor asignado a este parámetro puede influir notablemente en la precisión del modelo de identificación.

De aquí surge, por tanto, un problema de optimización: encontrar el valor óptimo del parámetro  $\lambda$ . Este valor óptimo sería responsable de que el modelo de identificación de sistemas correspondiente permitiese obtener un valor mínimo de la función de coste o *fitness*.

La función de *fitness*, en este caso, consiste en la suma acumulada de todos los errores cometidos en la estimación de todas las muestras de la serie temporal durante su identificación, tal como se expresa en la ecuación (22). En esta ecuación,  $NM$  es el número de muestras total de la serie temporal y  $k_0$  el instante inicial a partir de la cual se hace la identificación.

Como se ha dicho, la IS se basa en modelos construidos mediante el algoritmo RLS, donde es clave el valor que pueda tomar el parámetro  $\lambda$ . Por tanto, la correcta selección de su valor influye en un menor o mayor error cometido en la estimación. Así, el problema de optimización queda definido como la búsqueda de un valor óptimo de  $\lambda$  que minimice el error en la estimación, problema que se ha abordado mediante AGs.

$$(22) \quad f(\lambda) = \sum_{k=k_0}^{k=k_0+NM-1} |y_e(k) - y(k)|$$

#### 3.4.1 Identificación de sistemas en tiempo real

La IS encuentra modelos matemáticos de series complejas cuya  $y(k)$  es muestreada en el tiempo mediante un sensor que mide los datos generados. Generalmente, el modelo es generado “a posteriori”, utilizando los datos observados y almacenados.

No obstante, el interés de este trabajo se desplaza al caso de que existan series cuyo comportamiento se quiere predecir en tiempo de ejecución, esto es, mientras el sistema está en proceso y su  $y(k)$  está siendo observada. Este puede ser el caso de algún proceso de precisión cuya señal de salida es necesario mantener dentro de ciertos límites, debido a su comportamiento rápido e impredecible.

De esta forma, sería interesante poder generar modelos matemáticos al mismo tiempo que se ejecuta la identificación, de forma que, por ejemplo, un microprocesador pueda ir simulando el comportamiento próximo del sistema.

Por estas razones, se propone la definición de **Identificación de Sistemas en Tiempo Real (ISTR)** al proceso de IS que utiliza los algoritmos RLS al mismo tiempo que se muestrea la serie, de forma que se pueda construir un modelo matemático, basado en ARMAX, más perfecto cuanto más avance el tiempo.

### 3.4.2 Influencia del forgetting factor en la precisión de la identificación

La constante *forgetting-factor*  $\lambda$  tiene una gran influencia en la precisión; esta es la razón por la que se ha escogido como parámetro a optimizar.

Esta influencia puede ilustrarse por medio del siguiente ejemplo. Sea el *benchmark ball*, modelado con las dimensiones ARMAX  $na=3$ ,  $nb=2$ ,  $nc=1$ ,  $nk=1$ . Para los valores de  $ff$  0,7, 1,03, 1,36 y 1,7 obsérvese en la Figura 14 cómo la salida estimada es más o menos perfecta según el valor escogido, es decir, cómo afecta la sensibilidad de  $ff$  en la identificación.

Evidentemente, es necesario encontrar un buen valor de  $ff$ . A primera vista, se observa que los dos últimos (1,36 y 1,7) son muy malos.

Tras hacer un cómputo intensivo de 155 valores de  $ff$ , en el rango entre 0,6 y 1,2, se puede observar en la Figura 15 el valor de  $F$  para cada  $ff$ , y el valor óptimo encontrado ( $ff = 0,98961$  con  $F = 1,40762$ ). De aquí se deduce que, para hallar un valor óptimo, hay que hacer cálculos intensivos, cosa que en determinadas circunstancias (ISTR) podría ser difícil.

Por otro lado, no puede decirse que  $ff = 0,98961$  sea, definitivamente, el valor óptimo para la constante del *forgetting-factor*: lo es, pero solo para el *benchmark ball* y para el modelo ARMAX con unas dimensiones determinadas. Así, aplicando este  $ff$  a otros *benchmarks* (con las mismas dimensiones del modelo) obtenemos los resultados que se muestran en Tabla 3.

De la información mostrada en esta tabla se deduce que cada sistema tiene sus propios valores óptimos. Está claro que, si se utiliza un valor determinado de  $ff$  (por ejemplo, 0,95) como valor general para identificar toda clase de series temporales, se puede estar perdiendo precisión en la identificación de muchas de ellas. Vale la pena, pues, idear algoritmos que permitan obtener buenos valores de  $ff$  para cada sistema en particular.

En la Figura 16 se observa mejor la misma información de la Tabla 3 en cuanto a la diferencia entre el  $F$  encontrado para el valor óptimo de  $ff$ , con su valor correspondiente a  $ff = 0,98941$ .

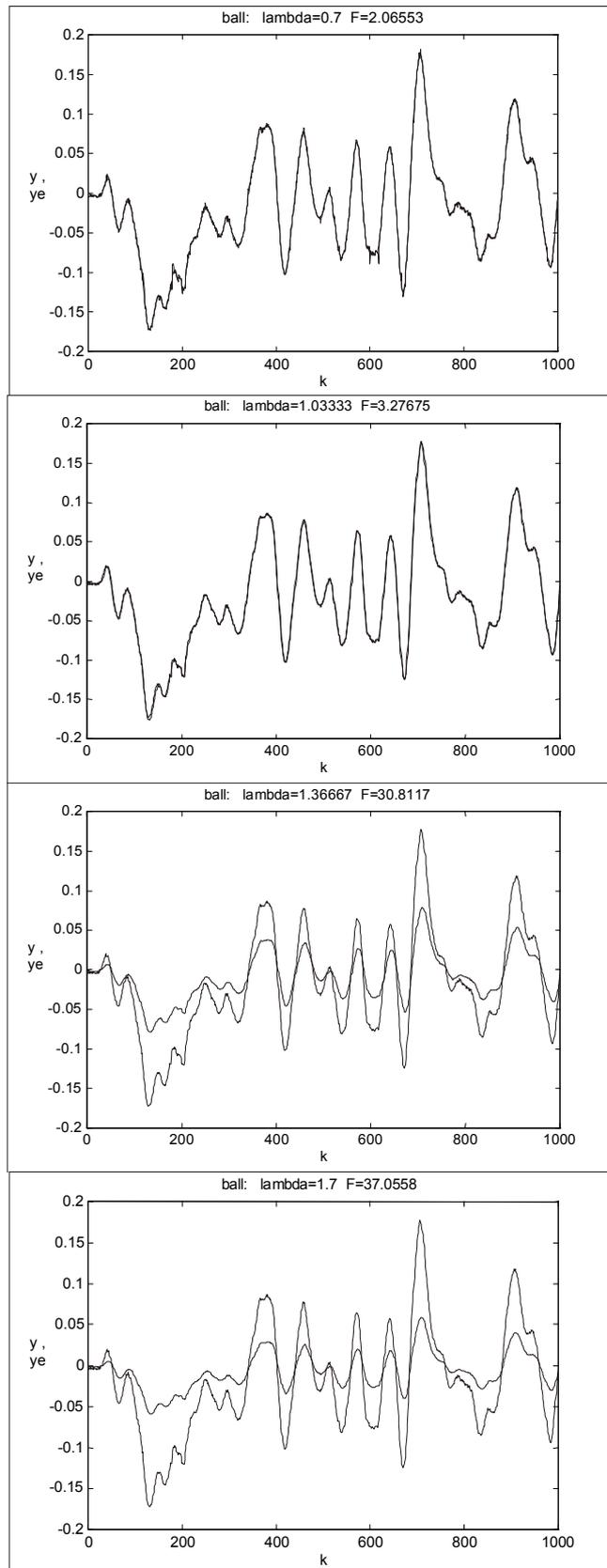


Figura 14. Influencia de *forgetting-factor* en la precisión de la identificación.

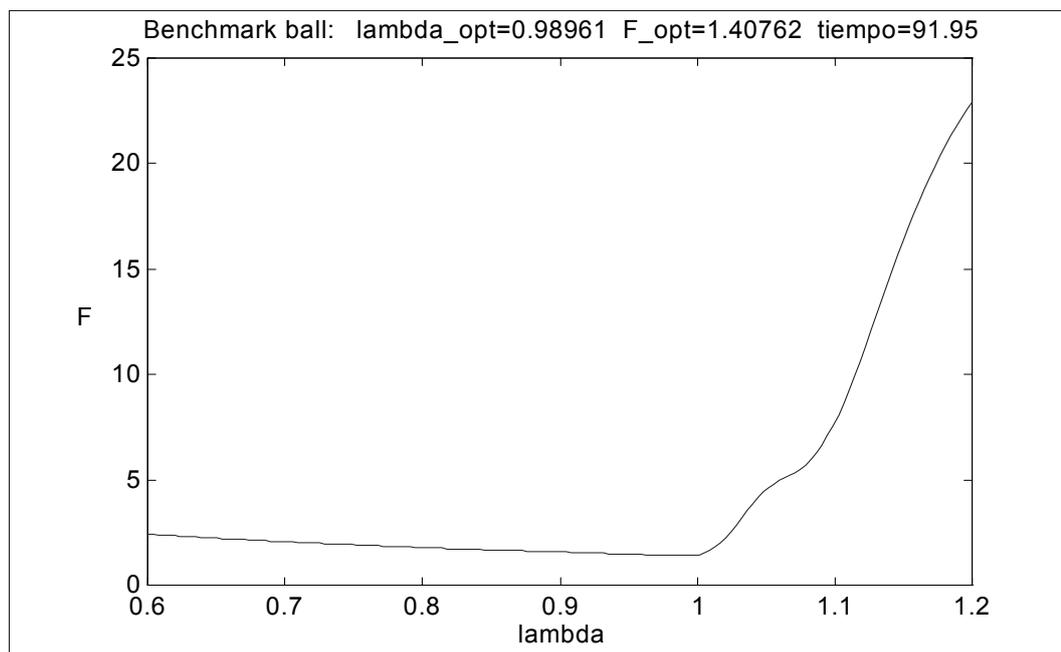


Figura 15. Fitness frente al parámetro forgetting-factor a optimizar. El mínimo de la curva corresponde con el valor óptimo.

benchmark	$\lambda$ óptimo (*)	F óptimo	F para $\lambda=0,98961$	Diferencia con F óptimo (%)	Rango utilizado $\lambda_c == R == NI$ (**)	mínimo $\lambda$ (***)
boxjen	1,0008	133,9	134,14	0,17 %	1,5 == 2,4 == 501	
cstr1	0,94711	0,49169	0,56461	14,83 %	0,95 == 0,078 == 55	0,911
cstr2	1,0002	518,1	520,91	0,54 %	0,96 == 0,0984 == 99	0,911
dryer	1,012	74,484	75,864	1,85 %	1 == 0,1 == 201	
exch	1,0288	1,435	1.456,8	101419,16 %	1 == 0,32 == 201	0,84
heat1	0,947	316,97	322,45	1,72 %	0,95 == 0,05 == 201	
robarm (*)	0,51	8,2169	11,193	36,21 %	1 == 0,98 == 201	0,5
fluttr (*)	0,51	0,7376	7,0846	860,49 %	1 == 0,98 == 201	0,5
gen0	1,015	0,26633	0,30259	13,61 %	0,9 == 0,3 == 201	
gen1	1,0625	43,727	58,718	34,28 %	1 == 0,5 == 201	

(\*) En estos casos, el  $\lambda$  óptimo es el más pequeño que se puede utilizar mientras el cómputo matricial sea consistente.

(\*\*) Los valores de  $\lambda$  se generan tomando uno central,  $\lambda_c$ , y definiendo un radio de generación, R, de forma que el rango es  $\{\lambda_c-R/2, \dots, \lambda_c+R/2\}$ , del que se toman valores de  $\lambda_c$  para hacer las NI identificaciones.

(\*\*\*) Umbral mínimo posible.

Tabla 3. Para cada benchmark, se compara la función de coste F obtenida para  $ff=0,9896$ , con el F óptimo calculado a partir de un  $ff$  encontrado de entre un rango, estableciendo la diferencia entre ambos F en forma de porcentaje.

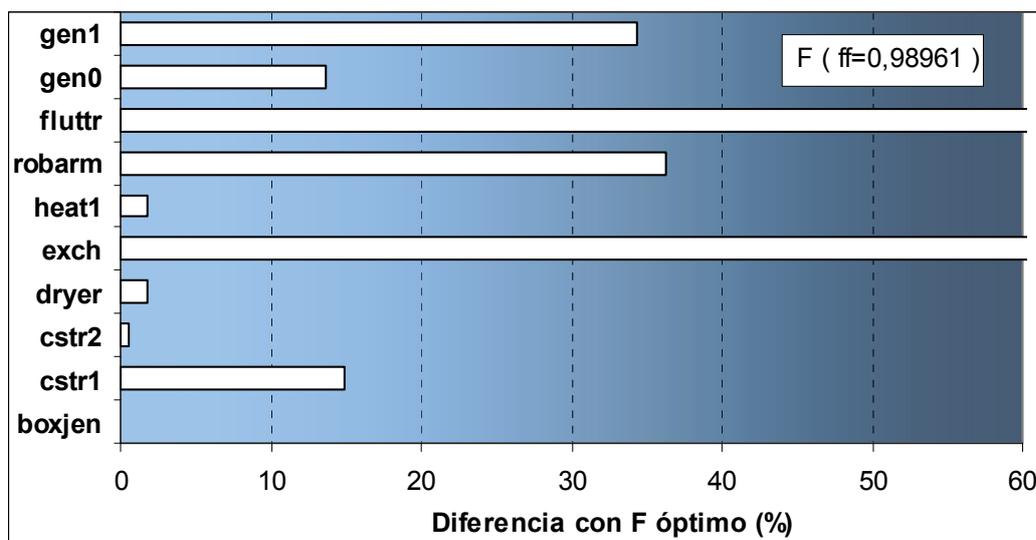


Figura 16. Diferencia con el F óptimo tal como se indica en la Tabla 3.

Hay bastante diferencia entre las distintas series temporales. Esto se debe a que el valor mínimo del *fitness* puede diferir mucho de los correspondientes a valores de *ff* más cercanos al óptimo. Esto se puede apreciar mejor en la Figura 17, donde se observa que no todas las series tienen el mismo comportamiento a la hora de ser identificados. Además, conviene hacer dos observaciones importantes:

- El algoritmo RLS puede hacer que, en algunas series y tras cierto número de muestras, algunos valores de sus matrices crezcan tanto (hacia  $\pm\infty$ ) que no puedan ser tratadas numéricamente. Por ejemplo, para los *benchmarks* *cstr1* y *cstr2*, no se pueden utilizar *ff* inferiores a 0,911. A estos valores de *ff* se les denominan como "umbrales mínimos".
- Hay series (*robarm* y *flutter*) cuyos *ff* óptimos son los más pequeños posibles.

Estas dos consideraciones serán de gran importancia a la hora de diseñar metodologías computacionales que sean precisas y eficientes.

### 3.4.3 Selección del tamaño óptimo de los modelos

Una consideración previa a la aplicación de los algoritmos genéticos para optimizar el valor de  $\lambda$  es determinar el tamaño más adecuado de los modelos a generar en la identificación, es decir, el grado (o dimensión) del polinomio ARMAX que representa la formulación matemática de la serie temporal: *na*.

Por tanto, para la identificación de sistemas es necesario seleccionar:

- El tamaño del modelo (*na*).
- El valor forgetting-factor ( $\lambda$ ).

En la Figura 20 se recoge el resultado del análisis efectuado para algunos ejemplos de *benchmarks* de series temporales, donde se puede observarse la evolución de la función de coste  $F$ .

Para cada *benchmark*, se ha efectuado el siguiente procedimiento experimental:

- Se han evaluado los resultados de  $F$  para 200 valores de  $ff$  escogidos, a intervalos regulares, del rango  $\{0.9 - 1.1\}$ .
- El mejor de estos resultados obtenidos es denominado  $F_{opt}$ , y se anota para el valor de la dimensión considerada del modelo,  $na$ .
- Este análisis se repite para 147 valores de  $na$  distintos, comprendidos entre 3 y 150. Los  $F_{opt}$  así generados son representados gráficamente en la Figura 20.
- Del análisis de esta gráfica se deduce que no se puede hablar de una pauta general que permita determinar exactamente cuál es la mejor dimensión de los modelos. Sin embargo, sí se constata una **tendencia** que empeora los resultados si el tamaño del modelo aumenta cada vez más. A partir de los datos aquí obtenidos, junto con las conclusiones extraídas de las experimentaciones efectuadas en otros trabajos, se puede fijar un valor adecuado de la dimensión de los modelos en  $na=15$ .
- Una conclusión operativa para el problema de optimización es que, si se utiliza un modelo de dimensión aceptable (es decir, que no se aleje de los resultados óptimos en un tiempo razonable), se podrá concluir que  $ff$  es el parámetro de optimización más importante a tener en cuenta.

#### **3.4.4 Influencia del muestreo de la serie temporal**

La influencia de las características del muestreo de la serie temporal en la identificación de sistemas es patente, pero depende de la forma en que el suministrador de la serie temporal la haya muestreado, pudiendo limitar la eficacia de las metodologías de IS. Estas características son dos:

- **Periodo de muestreo:** Si una serie temporal tiene una rápida variabilidad en su comportamiento, cuanto mayor sea el periodo  $T$  con el que se ha muestreado, peor se hará la identificación, pues se pierde información valiosa del comportamiento de la serie, información necesaria para determinar la naturaleza del sistema, tal como se puede apreciar en la Figura 18.
- **Número de muestras:** Si hay pocas muestras en relación con la historia completa de la serie, se resta información que pudiera contribuir para la obtención de un buen modelo matemático (Figura 19).

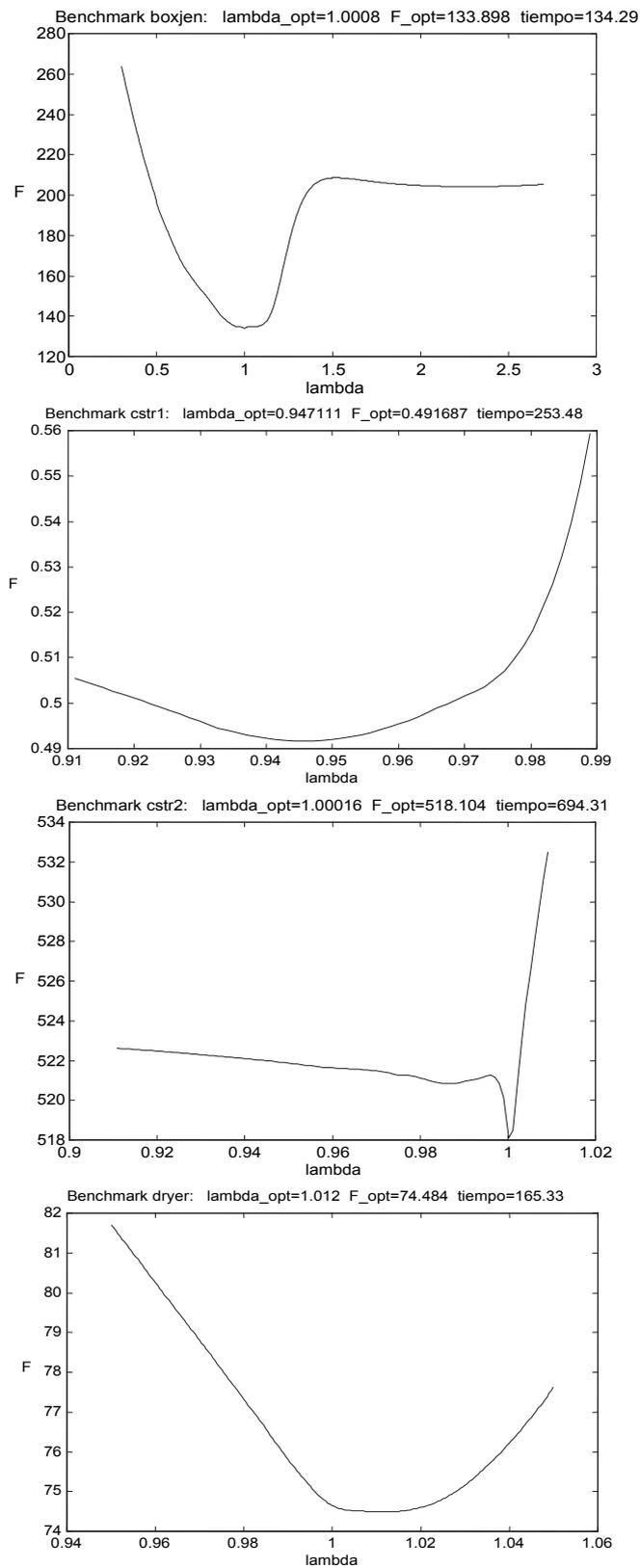


Figura 17. Valor de la función de fitness  $F$  frente a  $\lambda$ , para todos los *benchmarks* y rangos especificados en la Tabla 3. Obsérvese el mayor o menor pronunciamiento de sus valores mínimos ( $F_{opt}$ ).

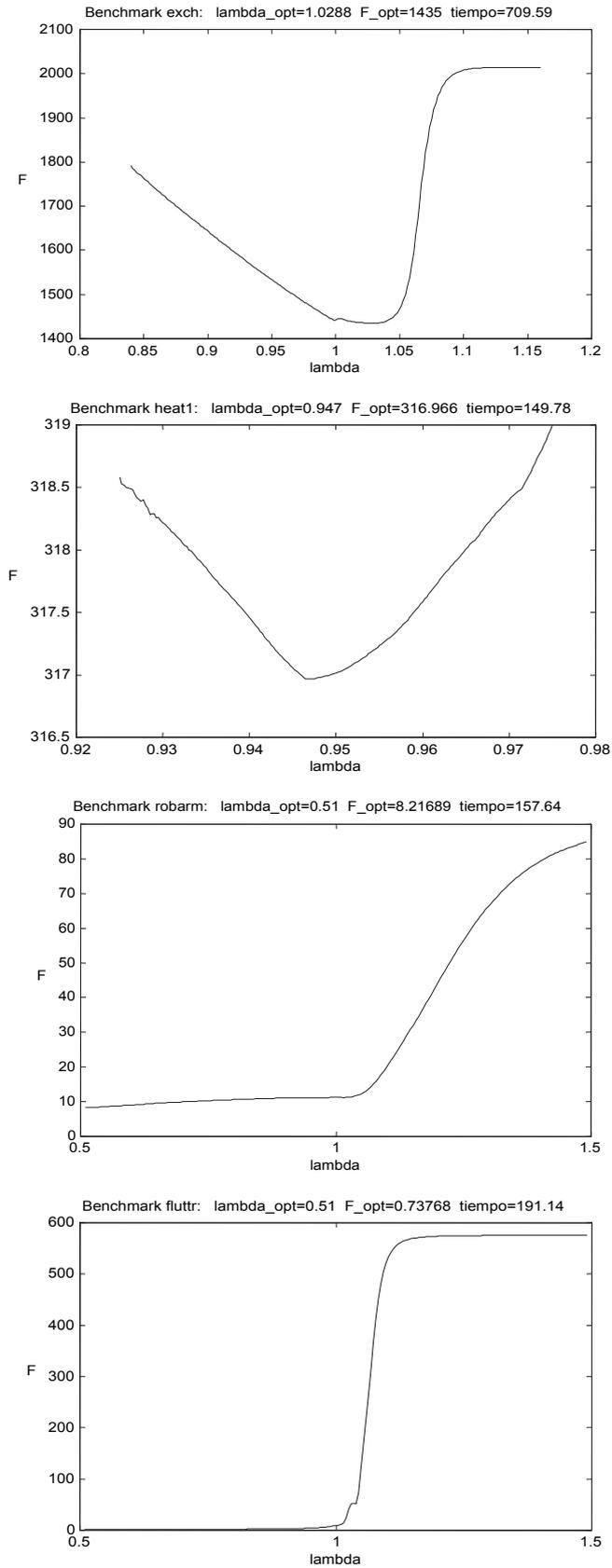


Figura 17. (continuación).

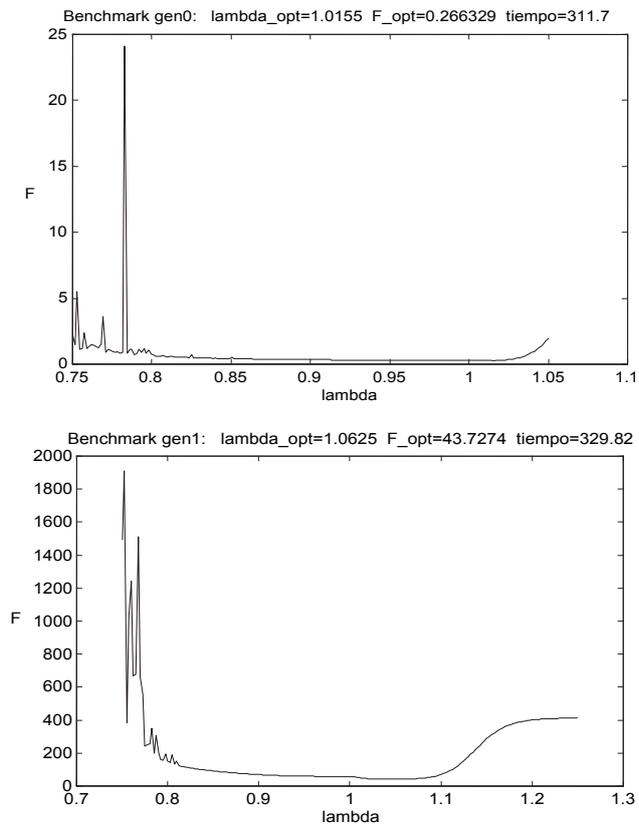


Figura 17. (continuación).

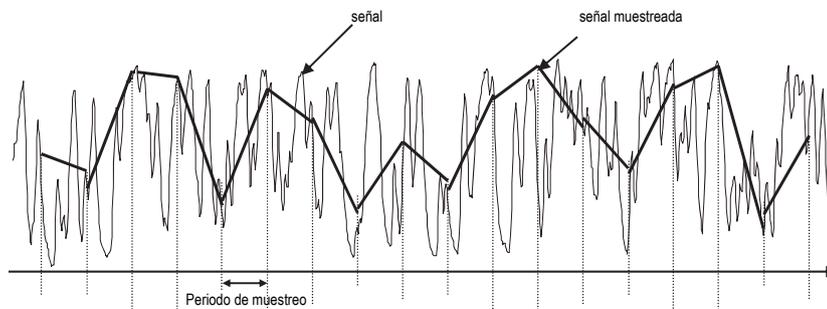


Figura 18. El periodo de muestreo considerado influye en la diferencia entre la serie muestreada y la serie real.

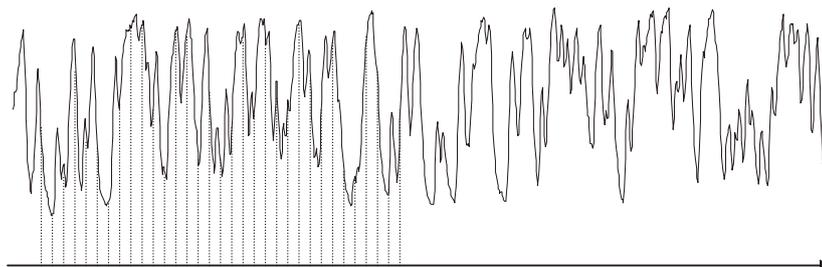


Figura 19. El nº de muestras consideradas influye en la precisión del modelo real.

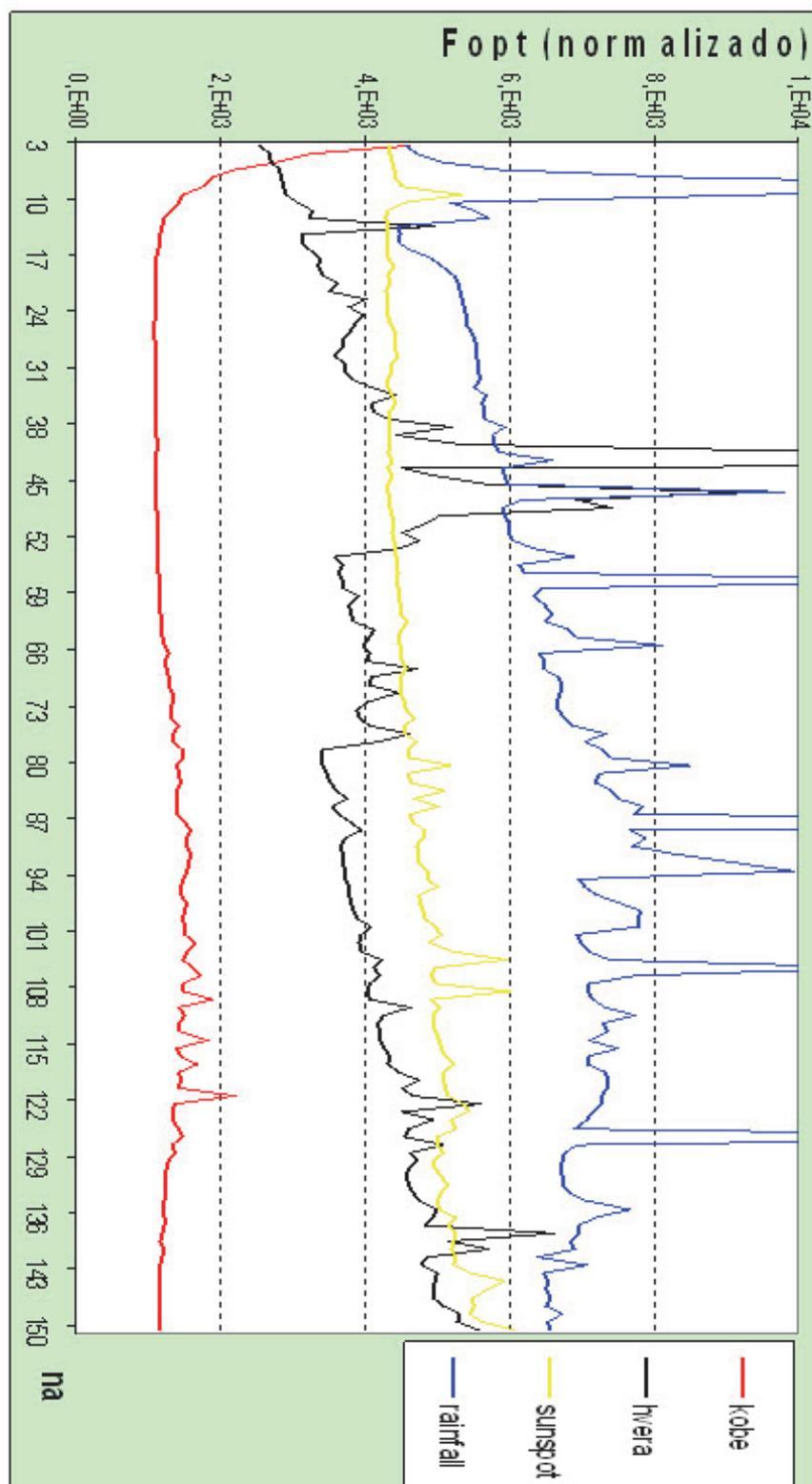


Figura 20. Mejor error cometido en la estimación para 200 valores de  $\lambda$  ( $F_{opt}$ ) frente al tamaño del modelo considerado (polinomio ARMAX desde grado 3 hasta 150),  $na$ . Análisis hecho para cuatro *benchmarks* de series temporales de distintos tamaños.

### 3.5 **Benchmarks utilizados**

Se conoce por *benchmark* a un valor de referencia que se utiliza en computación. En este trabajo, se trata de una serie temporal de prueba con la cual poder validar las metodologías y algoritmos desarrollados.

Se han considerado 12 *benchmarks* de series temporales de distintos tamaños (número de muestras), obtenidos de diversas bases de datos accesibles públicamente [TS 2002][TS 2003][TS 2004].

Cada *benchmark* viene especificado por un fichero de texto ASCII, *file\_s.dat*, que contiene números dispuestos en una columna, representando los valores de la señal en cada instante de muestreo.

Los *benchmarks* de TS utilizados para realizar este trabajo fueron:

- Ball.ts
- Eeg01.ts
- Eeg02.ts
- Eeg03.ts
- Eeg04.ts
- Hvera.ts
- Kobe.ts
- Melbmax.ts
- Melbmin.ts
- Rainfall.ts
- Sunspot.ts
- Z002.ts

### **3.6 Algoritmo genético paralelo para la optimización de ff**

Como ya se ha comentado anteriormente, se ha escogido la constante *forgetting-factor* como parámetro a optimizar ya que tiene una gran influencia en la precisión de los modelos obtenidos a partir de la IS. Por tanto, cada valor de  $\lambda$  será un individuo en el algoritmo genético.

#### **3.6.1 Especificaciones del algoritmo**

En este trabajo de optimización se ha querido aplicar tanto la formulación clásica del AG como una pequeña adaptación del mismo para hacer tareas paralelas. A este algoritmo adaptado se le llama **Algoritmo Genético Paralelo (AGP)**.

El AGP que se ha programado puede encuadrarse dentro de la categoría de algoritmos genéticos paralelos de grano grueso. Este tipo de algoritmos utiliza un conjunto de sub-poblaciones que evolucionan de forma individual respecto de las demás pero que, tras un número determinado de generaciones, intercambian sus individuos mutuamente con el objeto de introducir más variedad en la evolución.

En la Figura 21 puede observarse el esquema del AGP considerado, en el que existen 5 algoritmos genéticos simples que evolucionan en paralelo de forma independiente en cada fase. Una fase es una sucesión de un número determinado de generaciones (en dicha figura, aparecen 3 generaciones por fase). Al cabo de cada fase se produce una reasignación de las poblaciones finales de cada algoritmo genético, de forma que los individuos de la población final del algoritmo genético  $i$  pasan (sustituyen) a los individuos de la población final del algoritmo genético  $i+1$ , estableciéndose así una comunicación en forma de anillo.

Esta reasignación de poblaciones permite introducir más variedad en la evolución de los AGs. Pero esta variedad no se limita a dicha reasignación. Así, se ha programado que cada AG evolucione con distintas políticas de mutación.

Se considera como valor final óptimo de  $\lambda$  (solución del problema de optimización) aquél de los individuos existentes en las poblaciones finales de la última fase del AGP, para el cual la función de coste es la mínima.

Para todos los AGs que componen el AGP se ha considerado el mismo tamaño de población, políticas de selección y cruce. En este trabajo se ha implementado una sola versión del AGP, de las siguientes características:

- 5 algoritmos genéticos.
- 4 fases.
- 3 generaciones por fase.
- 20 individuos en cada población.

La razón de escoger estos valores estriba en el equilibrio a conseguir entre:

- Valores no muy bajos de los parámetros.
- Coste computacional en tiempo no muy alto.

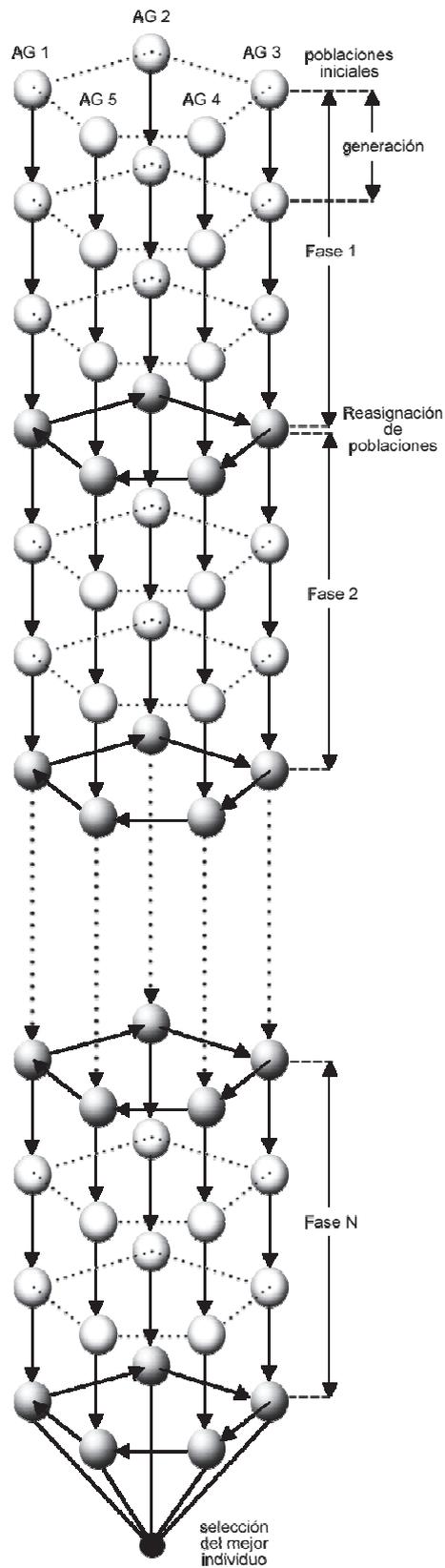


Figura 21. Esquema del algoritmo genético paralelo considerado en el problema.

### 3.6.2 Herramientas utilizadas

Como plataforma de cálculo se utilizó un PC Pentium IV a 3 GHz (año 2004). En este hardware, una ejecución del AGP con los valores descritos en la sección anterior puede tardar unas 24 horas para una serie temporal de 4.000 muestras.

Para la implementación del algoritmo genético paralelo se utilizó el lenguaje de programación Matlab. Este es un lenguaje de programación de alto nivel, interpretado, ideado para el desarrollo de aplicaciones en diversos campos de la Ingeniería dada la facilidad para expresar las operaciones en notación matemática y con operadores matriciales, y con numerosas librerías especializadas.

La razón de optar por este lenguaje de programación se fundamenta en las abundantes operaciones aritméticas matriciales, como puede observarse en las ecuaciones que definen el modelo ARMAX y el algoritmo RLS. La abundancia de estas operaciones y su integración en los algoritmos desarrollados ha llevado a considerar este lenguaje de programación como plataforma de prototipado rápido, pues se fijó como objetivo validar las metodologías más que conseguir tiempos de computación mínimos (al validar los algoritmos que proporcionan precisión en la IS, éstos pueden ser programados en otros lenguajes para acelerar su cómputo).

MATLAB proporciona una librería (*Toolbox*) específica que ofrece herramientas para construir y configurar algoritmos genéticos a medida de las necesidades del programador. Esta librería se llama *Genetic Algorithm and Direct Search Toolbox*, GADST [Matlab 2012], y contiene una colección de funciones que extiende las capacidades de su herramienta de optimización (*Optimization Toolbox*) y del entorno de computación numérico. Incluye rutinas (archivos .m) para resolver problemas de optimización usando AGs y búsquedas directas (en la Tabla 4 se muestran las más importantes).

Para los propósitos de este trabajo de tesis, se han extendido las capacidades de esta *toolbox* mediante la escritura de archivos .m propios.

Función	Descripción
ga	Encuentra el mínimo de una función usando el algoritmo genético
gaoptimget	Obtiene valores de una estructura de opciones de un algoritmo genético
gaoptimset	Crea una estructura de opciones de algoritmo genético
gatool	Abre la herramienta Genetic Algorithm Tool

Tabla 4. Principales funciones de la toolbox GADST para operar con AGs.

En la metodología de uso de esta librería, se debe primero programar un código MATLAB en un fichero .m, el cual debe contener la función que se quiere optimizar. Dicho fichero de código fuente deberá aceptar un vector fila (*row vector*), cuya longitud es el número de variables independientes para la función de *fitness*, y devuelve un escalar con el valor de dicha función.

### 3.7 Resultados experimentales

En la **Tabla 5** se muestran los resultados experimentales obtenidos para una colección de 12 *benchmarks* de series temporales de distintos tamaños.

Los resultados seleccionados en esta tabla son los valores de la función de *fitness* correspondientes a los valores óptimos del *forgetting-factor*, es decir, los valores mínimos de la función de *fitness*.

Se han comparado estos valores para las dos implementaciones, AG y AGP, observando que funciona ligeramente mejor, en la mayoría de los casos, la implementación paralela.

El AGP ha sido simulado en un entorno monoprocesador de forma concurrente.

Serie	AG: Fopt	AGP: Fopt	% AGP mejor AG
kobe	1.15633E+06	1.15507E+06	<b>0.11%</b>
s059	1.12324E+05	1.11796E+05	<b>0.47%</b>
sunspot	3.02724E+04	3.02716E+04	<b>0.003%</b>
rainfall	6.69450E+04	6.69417E+04	<b>0.005%</b>
melbmin	7.65593E+03	7.65535E+03	<b>0.008%</b>
melbmax	1.09016E+04	1.08970E+04	<b>0.04%</b>
tnzcl	1.04124E+02	1.04111E+02	<b>0.01%</b>
soi	7.38058E+03	7.38192E+03	<b>-0.02%</b>
hvera	3.13591E+03	3.13554E+03	<b>0.01%</b>
arch	1.10792E+03	1.10794E+03	<b>-0.002%</b>
n005	9.40755E+04	9.40272E+04	<b>0.05%</b>
z002	3.52604E+04	3.49646E+04	<b>0.84%</b>

Tabla 5. Comparativa de rendimientos AGP vs. AG, entendiendo por rendimiento el valor mínimo del *fitness*. Añadimos a la comparativa el porcentaje de cuánto más pequeño es el *fitness* óptimo del AGP respecto del AG.

El algoritmo genético simple tiene la siguiente configuración:

- Población: 20 individuos.
- Número generaciones: 12.

La razón de elegir estos valores ha sido el de obtener una solución óptima en un tiempo razonable para teniendo en cuenta el hardware utilizado.

El algoritmo genético paralelo tiene la siguiente configuración:

- Población: 20 individuos.
- Número fases: 4.
- Número de generaciones por fase: 3.
- Número de unidades de proceso (AGs) paralelos: 5.

Para cada serie temporal se ha aplicado el AGP y se ha consignado el mejor individuo encontrado de entre las poblaciones finales. Este resultado se ha comparado con el obtenido a partir de la ejecución de un AG simple, con propósitos de evaluación del rendimiento del AGP.

Para que esta comparación sea creíble, el AG tiene el mismo coste computacional en tiempo que cualquiera de los AGs de que consta el AGP (es como si éstos fueran ejecutados en máquinas paralelas). En otras palabras, se considera el mismo número de generaciones, tamaño de población, políticas de selección y cruce para ambos algoritmos.

En cambio, ambos algoritmos se diferencian en cuanto a su política de mutación. La mutación en el AG simple es la versión más común de las distintas versiones de las políticas de mutación utilizadas por los AGs que componen el AGP. La función de mutación utilizada añade un número aleatorio tomado a partir de una distribución gaussiana a cada entrada del vector padre. La varianza de esta distribución es determinada por dos parámetros, cuyos valores son los que determinan las diferentes versiones de las políticas de mutación consideradas en los algoritmos desarrollados.

Se han realizado varias ejecuciones para cada experimento con el AG y el AGP, con el fin de establecer los resultados mostrados en la **Tabla 5**.

### **3.8 Conclusiones**

Del análisis de los resultados experimentales se constata una efectiva mejoría del algoritmo genético paralelo frente al algoritmo genético simple, para la mayoría de los casos. No obstante, la mejora de los resultados es mínima, ya que el algoritmo genético simple ya ofrece muy buenos resultados por sí mismo.

Hay que precisar que, si bien puede parecer una mejoría mínima, en muchos casos la precisión requerida a los modelos es esencial, sobre todo para series temporales de sistemas cuyo tiempo de muestreo es muy pequeño o cuyas necesidades de predicción sean críticas por cuestiones de seguridad (procesos industriales, químicos, etc.), tal como ya se expuso anteriormente al hablar de la identificación de sistemas en tiempo real.

Varias son las líneas de trabajo que pueden ampliar este estudio:

- Se puede realizar un exhaustivo estudio sintonizando los parámetros del AGP (variando el número de algoritmos que evolucionan en paralelo, el número de fases, el número de generaciones por fase, las políticas de selección y de cruce, etc.). Es posible que en alguna de estas nuevas versiones pudiera aumentar aún más el rendimiento del AGP frente al AG simple.
- También se pueden probar otros algoritmos evolutivos, si bien con AG y AGP se ha cubierto el objetivo inicial del problema planteado: que la optimización de los modelos de identificación de sistemas puede mejorarse de forma satisfactoria utilizando algoritmos evolutivos.

### 3.9 Aportaciones

Como aportaciones realizadas por la doctoranda en este problema de optimización de series temporales, se citan las siguientes:

- Definición del problema de optimización en cuanto a establecer el parámetro *forgetting-factor* como objetivo a optimizar, frente a otros posibles parámetros, como es la dimensión del modelo. Para establecer este parámetro como objetivo a optimizar, se tuvo que acometer numerosas experimentaciones, con el objetivo de verificar, para distintos *benchmarks* de series temporales, la sensibilidad de la identificación frente a la variabilidad del tamaño del modelo y del *forgetting-factor*.
- Búsqueda y establecimiento de umbrales límites a los posibles valores para el *forgetting-factor*, así como los valores de referencia para el tamaño de los modelos. Esta información es necesaria para la posterior utilización de cualquier tipo de algoritmo evolutivo.
- Propuesta de un algoritmo genético para resolver el problema de optimización.
- Propuesta de una implementación de algoritmo genético paralelo, ideado para mejorar los resultados del algoritmo genético simple.
- Integración del algoritmo genético en un sistema de experimentación remota para la identificación de series temporales vía web: MOSET y TIDESI.

En cuanto a la productividad científica, las anteriores aportaciones aparecen en las siguientes publicaciones:

- **Capítulos de libros.**
  - "Optimization of Time Series Using Parallel, Adaptive, and Neural Techniques", in: "Optimization Techniques for Solving Complex Problems". Juan A. Gómez Pulido, María D. Jaraíz Simón, Miguel A. Vega-Rodríguez, and Juan M. Sánchez Pérez. Wiley, 2009, pag:127-140. ISBN:978-0-470-29332-4.
- **Conferencias.**
  - "APRIS: Library and On-Line Tools to Drive Parallel Adaptive and Neural Heuristic to Improve System Identification Precision". Juan A. Gómez Pulido, Miguel A. Vega-Rodríguez, Juan M. Sánchez Pérez, María D. Jaraíz Simón, José M. Granado Criado. Proceedings of the 13th IEEE Mediterranean Electrotechnical Conference, IEEE, Benalmádena, Spain, 2006, pag:425-428. ISBN:1-4244-0088-0.
  - "MOSET y TIDESI: Servicios web para el estudio de la identificación de sistemas aplicada a series temporales". Juan Antonio Gómez Pulido, Juan Manuel Sánchez Pérez, Miguel Ángel. Vega Rodríguez, María Dolores Jaraíz Simón. Actas de las IV Jornadas de Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática, Thomson-Paraninfo, Granada, Spain, 2005, pag:101-106. ISBN:84-9732-451-X.

- "Un algoritmo genético paralelo para el aumento de la precisión de los modelos de predicción de series temporales". Juan A. Gómez Pulido, Miguel A. Vega Rodríguez, Francisco Fernández de Vega, Juan Manuel Sánchez Pérez, María Dolores Jaraíz Simón. IV Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, Thomson-Paraninfo, Granada, Spain, 2005, pag:889-894. ISBN:84-9732-467-6.

# ***4 Optimización del diseño de códigos correctores de errores en comunicaciones***

*“La inteligencia es la habilidad  
de adaptarse al cambio.”*

**Stephen Hawking.**  
Científico inglés (1942-).

## **4.1 Resumen**

Durante la transmisión de la información digital a través de un canal, por ejemplo en comunicaciones inalámbricas, redes ad-hoc o redes de sensores inalámbricos, se producen errores prácticamente inevitables debido a la presencia de ruido, interferencias, ecos y otros factores de naturaleza física. Por esta razón es necesario establecer procedimientos, si no para evitar los errores, por lo menos poder reconocer su presencia y, en lo posible, corregirlos.

Algunos sistemas de telecomunicaciones no pueden tolerar el coste que supone la repetición de un mensaje cuando se corrompe por el camino. En su lugar, el mensaje debe ser corregido de alguna forma en el destino. En estos casos es adecuado el uso de los llamados **Códigos Correctores de Errores (ECC)**, que aplicados a los sistemas de comunicación agregan redundancia al mensaje para controlar el error. Esta redundancia agregada permite detectar y/o corregir (en su caso) los errores ocurridos durante la transmisión de los bloques de datos.

Hay varios tipos de códigos correctores de errores, pero en este trabajo de tesis solamente se contemplan los ECC binarios (palabras formadas por bits) de bloques lineales (palabras de tamaños fijos).

El problema de optimización que trata este trabajo de tesis consiste en encontrar el código óptimo, dado un número fijo de palabras en el código y un tamaño de palabra también determinado. Este es un problema de optimización combinatoria NP-Completo, pues existe un muy elevado número de posibles soluciones al problema (códigos correctores) que hacen imposible el uso de cualquier técnica numérica de naturaleza exacta.

Por esta razón se consideró abordar el problema mediante algoritmos genéticos, utilizando una instancia experimental igual a la que algunos artículos publicados han abordado con anterioridad, con el objetivo de comprobar el grado de éxito de la implementación desarrollada en este trabajo para el algoritmo genético.

Se han efectuado numerosos experimentos con varias ejecuciones para cada uno, con instancias experimentales de menor tamaño que la instancia objetivo. Estos experimentos tenían el propósito de sintonizar los parámetros y las políticas de los operadores del AG, y arrojar conclusiones extrapolables a instancias de mayor tamaño, pues la instancia objetivo (comparable con la de otros trabajos), consume tiempos de computación muy altos que dificultan hacer muchos experimentos.

Los resultados obtenidos en este trabajo de tesis para la instancia objetivo superan a otras implementaciones de AGs y a otras heurísticas de naturaleza evolutiva, si bien no superan al algoritmo CHC, que ha obtenido el mejor resultado.

Con el objeto de facilitar el marco experimental para la resolución del problema de optimización, así como para proporcionar una plataforma abierta a otros investigadores, se desarrolló un servicio web que permitía utilizar la implementación del algoritmo genético para cualquier instancia experimental, y visualizar, exportar y analizar de forma remota los resultados obtenidos.

## 4.2 Introducción y trabajos relacionados

Existen sistemas de comunicaciones, como las comunicaciones punto a punto, donde repetir el envío de un mensaje que ha llegado corrupto al destino puede no ser un inconveniente, ni suponer un gasto excesivo. No obstante, existen otros sistemas de comunicaciones, como las comunicaciones en tiempo real, especialmente multipunto, que no pueden tolerar el reenvío de un mensaje que se ha corrompido en el camino (telefonía, televisión, radio digital, etc.).

En el caso de las comunicaciones multipunto, la información debe ser entregada con bajo retraso, por lo que se hace adecuado utilizar un ECC para evitar las sobrecargas que supondrían la retransmisión.

Hay muchos tipos de ECC. El estudio que se aborda aquí se centra en los ECC binarios de bloques lineales. Estos códigos están formados por un conjunto de palabras, cada una de las cuales es una cadena binaria de cierta longitud.

Los mensajes están compuestos por una secuencia de palabras. De esta forma, cuando se transmite un mensaje, las cadenas binarias que lo forman son enviadas a través del medio físico formando un flujo de bits que puede sufrir modificaciones conforme atraviesa el medio. Cuando un bit de una palabra ha cambiado en el camino, se dice que hay un error en esa palabra. Al otro lado (en el receptor) llegaría una palabra incorrecta [Dontas 1990].

Si se quiere garantizar que se puedan detectar hasta  $e$  errores, la llamada *distancia de Hamming* (que se define un poco más adelante) entre cualquier par de palabras debe ser al menos  $e+1$ . Por otro lado, si la *distancia de Hamming* entre cada par de palabras es mayor o igual a  $2e+1$ , las cadenas con  $e$  o menos errores pueden corregirse. La corrección se consigue asignando a la cadena incorrecta su palabra de código más cercana en el destino. Así que, cuanto más separadas estén las palabras del código, más errores pueden ser corregidos.

Dada la longitud de las palabras y el número de palabras de un código, **el problema de encontrar un código óptimo es NP-Completo.**

Las técnicas exactas para determinar el código óptimo pueden ser útiles para pequeñas instancias (mensajes de pequeño tamaño), pero para instancias más realistas son inviables debido a su alto tiempo de cómputo. Por esta razón, se deben utilizar técnicas heurísticas.

En el marco de este problema de optimización, se ha aplicado *Simulated Annealing* [Gamal 1987], Algoritmo Repulsivo y diversas implementaciones de Algoritmo Genético [Alba 2004 a][Alba 2004 b]. En esta tesis doctoral se propone una implementación mejorada (en cuanto a la sintonización de sus parámetros y selección de políticas para los operadores) de algoritmo genético que mejora los resultados obtenidos en estos últimos trabajos.

### 4.3 Códigos correctores de errores

Para la corrección de errores existe una amplia variedad de códigos, tales como los códigos lineales, cíclicos, etc. Los ECC que se consideran en este trabajo son los de bloques, que se caracterizan por los siguientes parámetros:

- La **aridad** del código ( $q$ ): Representa el número de símbolos distintos en el alfabeto usado en el código. Los códigos binarios, que son los más comunes, tienen aridad 2 y usan el alfabeto  $\{0,1\}$ .
- La **longitud** de un código ( $l$ ): Es el número de letras (del alfabeto) necesario para formar una **palabra (vector)** en el código, donde un código denota un conjunto de palabras usadas para transmitir un mensaje. En el caso de códigos binarios, la longitud de la palabra se denota con la letra  $n$ , y es el número de bits (0 ó 1) que tiene cada palabra del código.
- El **tamaño** de un código ( $M$ ): Es el número de palabras distintas en el código usado para componer el mensaje enviado.
- **Distancia** ( $d$ ) de un código: Es la mínima distancia de Hamming entre pares de palabras pertenecientes al código.
- La **distancia Hamming** entre dos vectores  $x$  e  $y$ , denotada por  $d(x,y)$ , es el número de posiciones en las cuales los vectores  $x$  e  $y$  difieren. En otras palabras, la distancia Hamming entre  $x$  e  $y$  denota el mínimo número de operaciones de sustitución que deben ser realizadas para convertir el vector  $x$  en el vector  $y$ .
- Un código  $C$  puede corregir hasta  $t$  errores si  $d(C) > (2t+1)$ , donde,  $d(C) = \min\{d(x,y) | x,y \in C, x \neq y\}$ .

#### 4.4 El problema del diseño óptimo de códigos correctores de errores

Un problema básico en la construcción de cualquier sistema de comunicación es encontrar un código corrector adecuado para transmitir mensajes a través de un canal con ruido de la forma más rápida y fiable posible. Como suele ocurrir en optimización combinatoria, hay que identificar qué objetivo u objetivos en conflicto han de optimizarse. En este trabajo de tesis se considera un único objetivo, que es encontrar palabras con longitud mínima para que el mensaje se pueda transmitir rápidamente.

No obstante, la distancia de Hamming entre palabras debe ser máxima para garantizar un alto nivel de corrección en el lado del receptor, lo cual sugiere incluir alta redundancia (es decir, palabras más largas). Cuanto menor sea la longitud de las palabras, menor será la mínima distancia de Hamming entre ellas.

La idea, por tanto, es la siguiente: si todas las palabras están separadas por  $d$  bits, cualquier modificación de hasta  $(d-1)/2$  bits en una palabra válida puede ser corregida tomando la palabra más cercana, como puede verse en la Figura 22.

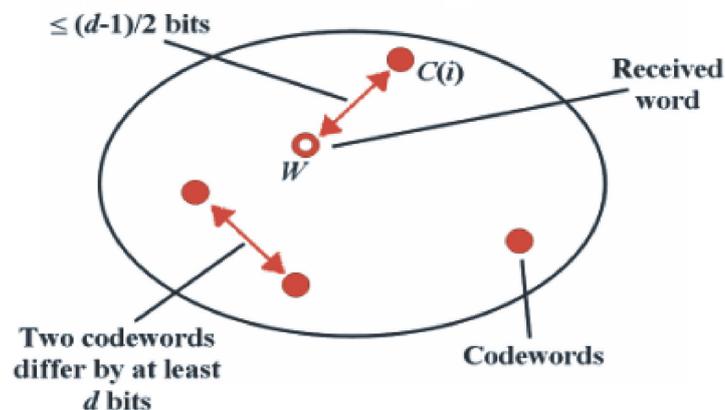


Figura 22. Interpretación gráfica de un sistema de código corrector de errores.

Los códigos binarios lineales de bloques pueden ser representados como un vector de tres parámetros  $(n, M, d)$ , donde  $n$  es el número de bits de cada palabra del código,  $M$  es el número de palabras y  $d$  es la mínima distancia de Hamming entre cualquier par de palabras  $C(i)$  y  $C(j)$  (donde  $i \neq j$ ). De esta forma, un **código óptimo** es aquel que maximiza  $d$ , para un valor dado de  $n$  y de  $M$ .

El trabajo de optimización que se aborda en esta tesis resuelve el problema para un código de 24 palabras y 12 bits ( $M=24, n=12$ ). A modo ilustrativo, el espacio de búsqueda es  $\binom{2^n}{M}$ , luego se generan aproximadamente  $10^{63}$  códigos diferentes,

los cuales tienen 24 palabras y 12 bits diferentes. Por tanto, la búsqueda exhaustiva de un código óptimo está claramente descartada.

## 4.5 Aplicación de un AG para la resolución del problema

El problema de optimización planteado en la sección anterior es abordado en este trabajo mediante un algoritmo genético.

### 4.5.1 Representación

Para codificar las soluciones del problema de optimización se usa una cadena binaria de longitud  $M \times n$ . El genotipo es la concatenación de las  $M$  palabras de longitud  $n$ . Ésta es una representación directa de los parámetros del problema en el cromosoma de cada individuo, proporcionando de esta forma una conversión sencilla entre el genotipo y el fenotipo.

En otras palabras, en este problema cada individuo de la población es un código binario. El cromosoma del individuo se representa mediante una matriz de  $M$  filas y  $n$  columnas. Cada elemento de la matriz es un valor binario (0 ó 1), que es generado de forma aleatoria. Por ejemplo, si una población está formada por 7 individuos para una instancia de tamaño  $\{M=4, n=3\}$ , una posible población sería la de la Figura 23.

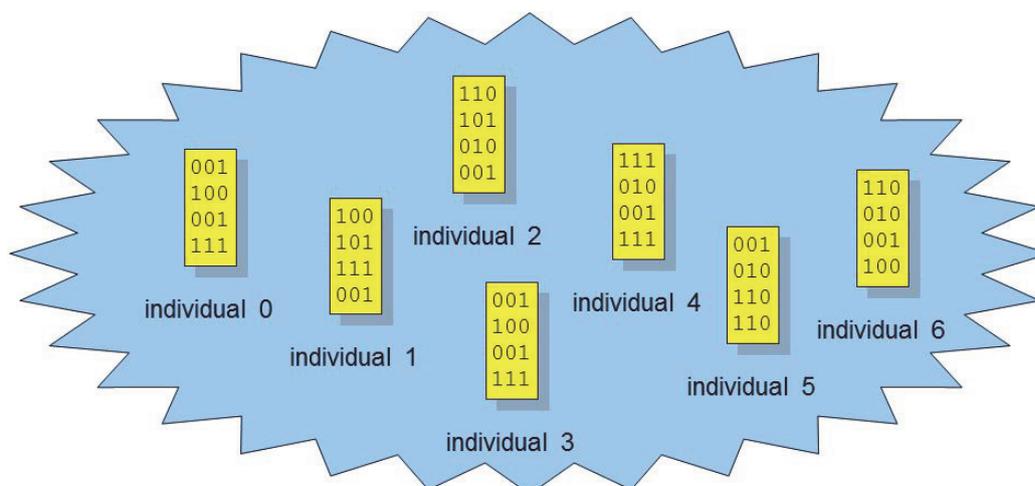


Figura 23. Población formada por 7 individuos. El cromosoma de cada individuo está representado por  $M$  filas y  $n$  columnas.

### 4.5.2 Función de fitness

Como función de *fitness* a maximizar, podría utilizarse la distancia mínima de Hamming entre las palabras del código, pero esta función es muy inexacta, ya que proporciona poca información acerca de las ventajas internas de dos soluciones que compiten. Una función de *fitness* más precisa es la propuesta en [Dontas 1990], cuya formulación se encuentra en (23), siendo  $x$  es el código (vector de palabras) y  $d_{ij}$  la distancia de Hamming entre  $x_i$  y  $x_j$ .

$$(23) \quad f(x) = \frac{1}{\sum_{i=1}^M \sum_{j=1, j \neq i}^M \frac{1}{d_{ij}^2}} = \frac{1}{2 \sum_{i=1}^{M-1} \sum_{j=i+1}^M d_{ij}^{-2}}$$

Esta función tiene un inconveniente: puede asignar un valor más alto de *fitness* a un código con menor distancia mínima de Hamming que otro. Esto se puede ver en el siguiente ejemplo: Sean  $C1$  y  $C2$  los siguientes dos códigos de diez bits y tres palabras:

$$C1 = \{0000000000, 0000011111, 0011100111\}$$

$$C2 = \{0000000000, 0000001111, 1111110011\}$$

La distancia mínima en el primero es  $d(C1) = 5$ , y en el segundo  $d(C2) = 4$ . No obstante, sus valores de *fitness* de acuerdo con (23) son  $f(C1)=4,639$  y  $f(C2)=5,333$ . Se puede observar que el código con mayor distancia mínima de Hamming entre palabras ( $C1$ ) tiene menor valor de *fitness* que el otro ( $C2$ ). Por tanto, para evitar este problema se ha añadido a la formulación dada en (23) un término que aumenta con respecto a la distancia mínima de Hamming.

Así, la nueva función de *fitness* considerada en este trabajo está descrita en (24), donde  $d_{min}$  es la mínima distancia de Hamming entre palabras del código.

$$(24) \quad f(x) = \frac{1}{\sum_{i=1}^M \sum_{j=1, j \neq i}^M \frac{1}{d_{ij}^2}} + \left( \frac{d_{min}}{12} - \frac{d_{min}^2}{4} + \frac{d_{min}^3}{6} \right)$$

La fórmula (24) contiene, por tanto, dos términos. El primero es exactamente la ecuación (23), que permite una distinción precisa entre códigos con la misma distancia mínima de Hamming pero diferente configuración de palabras. El segundo término garantiza que los valores de *fitness* de códigos con distancia mínima  $d$  sean menores que aquellos de códigos con distancia mínima  $d+1$ .

### 4.5.3 Características del algoritmo genético

Como se ha mencionado anteriormente, la optimización en esta tesis es abordada mediante un algoritmo genético, para el cual ha habido que hacer experimentaciones previas con el objeto de seleccionar valores adecuados para los distintos parámetros y políticas de sus operadores.

Una importante restricción en el trabajo desarrollado, en cuanto al esfuerzo de sintonización de los parámetros, ha sido el elevado tiempo de computación para la instancia experimental.

Esta instancia está determinada por el par de valores  $\{n=8, M=24\}$ , y es la que utilizan otros autores en artículos publicados [Alba 2004 a] [Alba 2004 b], lo cual permitió poder establecer comparativas de rendimiento.

Dados los elevados tiempos de computación obtenidos por el algoritmo genético para esta instancia experimental<sup>2</sup>, se estableció como valores razonables, según experimentaciones previas, los siguientes valores:

- Tamaño de población: 20 individuos.
- Criterio de parada: al alcanzar las 12 generaciones.

El conjunto de experimentaciones previas no solo sirvió para establecer el tamaño de población y el número de generaciones, sino también las políticas de los operadores genéticos. Para dar agilidad a la sintonización del AG, las experimentaciones previas se realizaron sobre instancias de menor coste computacional (menor valor de  $n$  y  $M$ ). Del análisis de los resultados se observó que era posible extrapolar las conclusiones para instancias de mayor tamaño.

De esta forma, se establecieron los siguientes valores:

- Operador de **selección**. Selección uniforme estocástica, traza una línea en la que cada padre corresponde a una sección de la línea de longitud proporcional a su valor.
- Operador de **cruce**. Mediante este operador se intercambian las filas de los padres para formar la descendencia.
- Operador de **mutación**. Se selecciona aleatoriamente un elemento (bit) de la matriz (código). Si el bit es 1 se cambia por 0 y viceversa. De esta forma, se consigue introducir cierto grado de variabilidad en los individuos de la descendencia.

---

<sup>2</sup> Las experimentaciones correspondientes a este trabajo se realizaron en el año 2005, disponiéndose entonces de un computador personal con procesador Pentium IV a 3 GHz.

## 4.6 Resultados experimentales

En este apartado se muestran los resultados experimentales obtenidos.

### 4.6.1 *Herramientas utilizadas*

El algoritmo genético fue programado mediante la librería *Genetic Algorithm and Direct Search Toolbox* del lenguaje de programación MATLAB [Matlab 2012], para la versión R14 (año 2005).

El hardware utilizado consistió en computadores personales con procesador Pentium IV a 3 GHz con 1 GB de memoria RAM.

Se utilizaron varios de estos computadores, sitos en un laboratorio de la Escuela Politécnica de la UEX, con el fin de poder efectuar varias ejecuciones de los mismos experimentos (algunos de estos experimentos se realizaron en épocas no lectivas, dados los elevados tiempos de computación necesarios).

### 4.6.2 *Instancias experimentales*

Como se ha comentado anteriormente, este trabajo considera una instancia determinada  $\{n=12, M=24\}$  con el objeto de poder comparar los resultados con otros publicados. A esta instancia se le da el nombre de **instancia objetivo**.

No obstante, se definió un conjunto de instancias de menor tamaño, llamadas **instancias experimentales**, con el objeto de realizar pruebas y sintonizaciones de parámetros y políticas de los operadores genéticos de una forma más ágil, dados los elevados tiempos de computación de la instancia objetivo. Las 15 instancias experimentales escogidas fueron las siguientes:

$\{n=8, M=8\}$ ,  $\{n=8, M=10\}$ ,  $\{n=8, M=12\}$ ,  $\{n=8, M=14\}$ ,  $\{n=8, M=16\}$ ,  
 $\{n=10, M=8\}$ ,  $\{n=10, M=10\}$ ,  $\{n=10, M=12\}$ ,  $\{n=10, M=14\}$ ,  $\{n=10, M=16\}$ ,  
 $\{n=12, M=8\}$ ,  $\{n=12, M=10\}$ ,  $\{n=12, M=12\}$ ,  $\{n=12, M=14\}$ ,  $\{n=12, M=16\}$ .

Nótese que la última instancia es la más cercana a la instancia objetivo, pero ofrece tiempos de computación mucho menor.

### 4.6.3 *Tiempos de computación*

Los tiempos de computación son muy sensibles al tamaño de la instancia experimental (para un mismo computador utilizado), tal como puede observarse en la Tabla 6. Así, a mayor tamaño de la instancia, mayor tiempo de computación. Tanto  $n$  como  $M$  incrementan el tiempo de computación; sin embargo,  $M$  incrementa el tiempo de computación en mayor cuantía.

#### 4.6.4 Ejecuciones

Se fijaron 10 ejecuciones del AG para cada experimento (teniendo en cuenta el tiempo de computación). No obstante, se constató que, en el caso de la instancia objetivo, todas las ejecuciones reportaban el mismo valor óptimo al cabo de las 12 generaciones. Por tanto, para la instancia objetivo y para la configuración fijada del algoritmo genético, no fue necesario realizar ningún tipo de análisis estadístico de los resultados.

Condiciones del AG	Instancia experimental	<i>Fitness</i> de la solución óptima	Tiempo de computación
Generaciones: 3 Población: 3 individuos	M=6 y n=6	1.632917	0,031 s
	M=24 y n=12	6.065413	202.070 s = 2 días
Generaciones: 12 Población: 24 individuos	M=24 y n=12	7.859838	1.176.022 s = 14 días

Tabla 6. Resultados experimentales del AG (una sola ejecución por experimento).

#### 4.6.5 Resultados con las instancias experimentales

Tal como puede observarse en la Tabla 6, el *fitness* mejora al aumentar el tamaño de la población y el número de generaciones, algo que es muy común en los problemas de optimización combinatoria abordados por AGs.

Esto se pone mejor de manifiesto en la Figura 24, donde se muestra la evolución de la solución óptima de las 15 instancias experimentales a lo largo de 150 generaciones, comparándose las gráficas según el tamaño del código corrector ( $M$ ), para distintos valores de la longitud de las palabras del código ( $n$ ).

Esta mejoría del *fitness* se produce, no obstante, en mayor o menor grado según el tamaño del código, esto es, según el número de palabras que hay en el código corrector de error. Así, se aprecia una convergencia hacia un valor estable para bajos valores de  $M$ , mientras que, cuantas más palabras tiene el código, mejores soluciones se encuentran a lo largo de las generaciones; esto es previsible, ya que el número de posibles soluciones óptimas aumenta cuantas más palabras hay en el código corrector de error (al mismo tiempo, se eleva considerablemente el coste computacional y el tiempo de ejecución).

La misma conclusión se extrae observando la Figura 25, que muestra la misma evolución de la solución óptima de las 15 instancias experimentales a lo largo de 150 generaciones, comparándose esta vez las gráficas según la longitud de las palabras del código ( $n$ ), para distintos tamaños de las palabras del código ( $M$ ). En esta figura se aprecia con más claridad el valor que adquiere la función de *fitness* para distintos valores de  $M$ : cuantas más palabras tiene el código, mayor es el valor de la función de *fitness*, para un mismo tamaño de la palabra del código.

Finalmente, en la Figura 26 se observa el valor del *fitness* de la solución óptima de las 15 instancias experimentales, para las primeras 4 generaciones del algoritmo

genético, con el objeto de ver cómo el valor del *fitness* mejora, por un lado, en las sucesivas generaciones y aumenta, por otro lado, cuantas más palabras tiene el código corrector.

La información que se extrae del análisis de estas gráficas se puede extrapolar a instancias de otros tamaños, ya que los aumentos de los valores de *fitness* son predecibles cualitativamente, según varían los valores de  $n$ ,  $M$  o el número de generaciones. Por tanto, estas conclusiones pueden extrapolarse al caso de la instancia objetivo. En esta instancia, el coste computacional crece considerablemente, al tener las palabras la longitud más alta contemplada (12 bits) y tener los códigos un número de palabras muy superior (24) al de todas las instancias experimentales consideradas.

#### **4.6.6 Comparativa de rendimiento con otros autores**

Los resultados de la instancia objetivo se han podido comparar con los obtenidos por otros autores que utilizaron diferentes técnicas heurísticas de optimización. El resultado del AG desarrollado en esta tesis se comparó con los obtenidos por otros autores [Alba 2004 a][Alba 2004 b], que utilizaron las siguientes heurísticas **con la misma instancia objetivo y la misma función de *fitness* adaptada**:

- **Algoritmo repulsivo (RA).**
- **AG de estado estacionario (ssGA).**
- **CHC** (*Cross generational elitist selection, Heterogeneous recombination and Cataclism mutation*, también conocido como AG con control de endogamia).
- **AG celular (cGA).**

Los resultados se muestran en la **Tabla 7**, donde se puede observar que la implementación del AG desarrollado en esta tesis ofrece **un excelente resultado**, solo superado por CHC, y ligeramente mejor que el obtenido por el AG celular.

Aquí hay que apuntar que las publicaciones donde aparecen estos resultados no informan del número de generaciones procesadas por los algoritmos genéticos, ya que utilizan como criterio de parada un número elevado ( $10^5$ ) de iteraciones entre todos los sub-algoritmos de que están compuestos.

Atendiendo a las consideraciones expuestas en el apartado 4.6.5, y dada la extrapolación de conclusiones fruto del análisis de las instancias experimentales, es más que probable que procesando unas pocas generaciones más del AG desarrollado, se pueda mejorar notablemente el valor del *fitness* óptimo, ya que precisamente con valores altos de  $M$  no se aprecia estancamiento en la mejoría del *fitness*; todo lo contrario, se encuentran cada vez mejores soluciones para sucesivas generaciones.

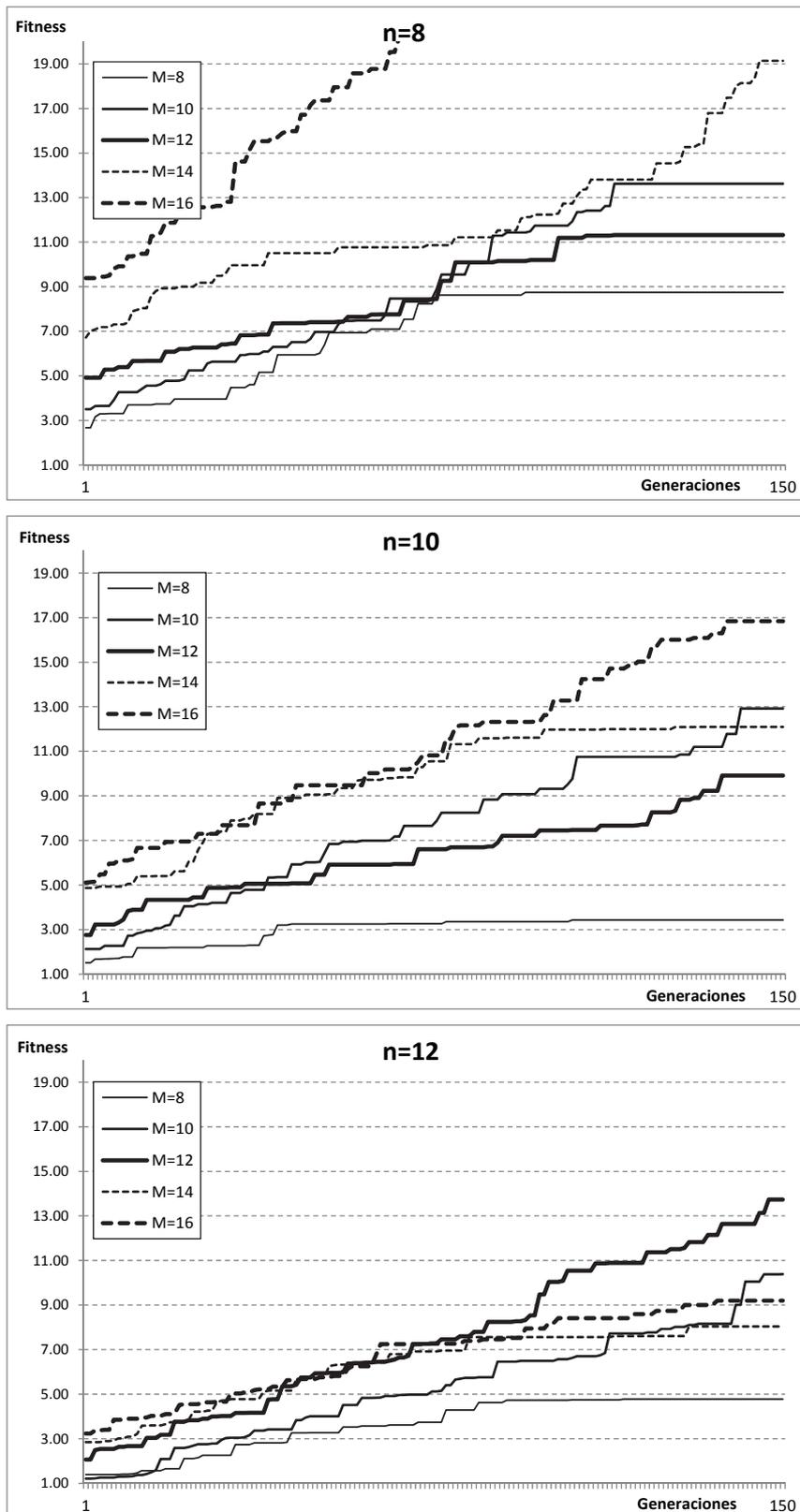


Figura 24. Evolución de la solución óptima de las 15 instancias experimentales utilizadas, a lo largo de 150 generaciones. Comparativa hecha según el tamaño del código corrector.

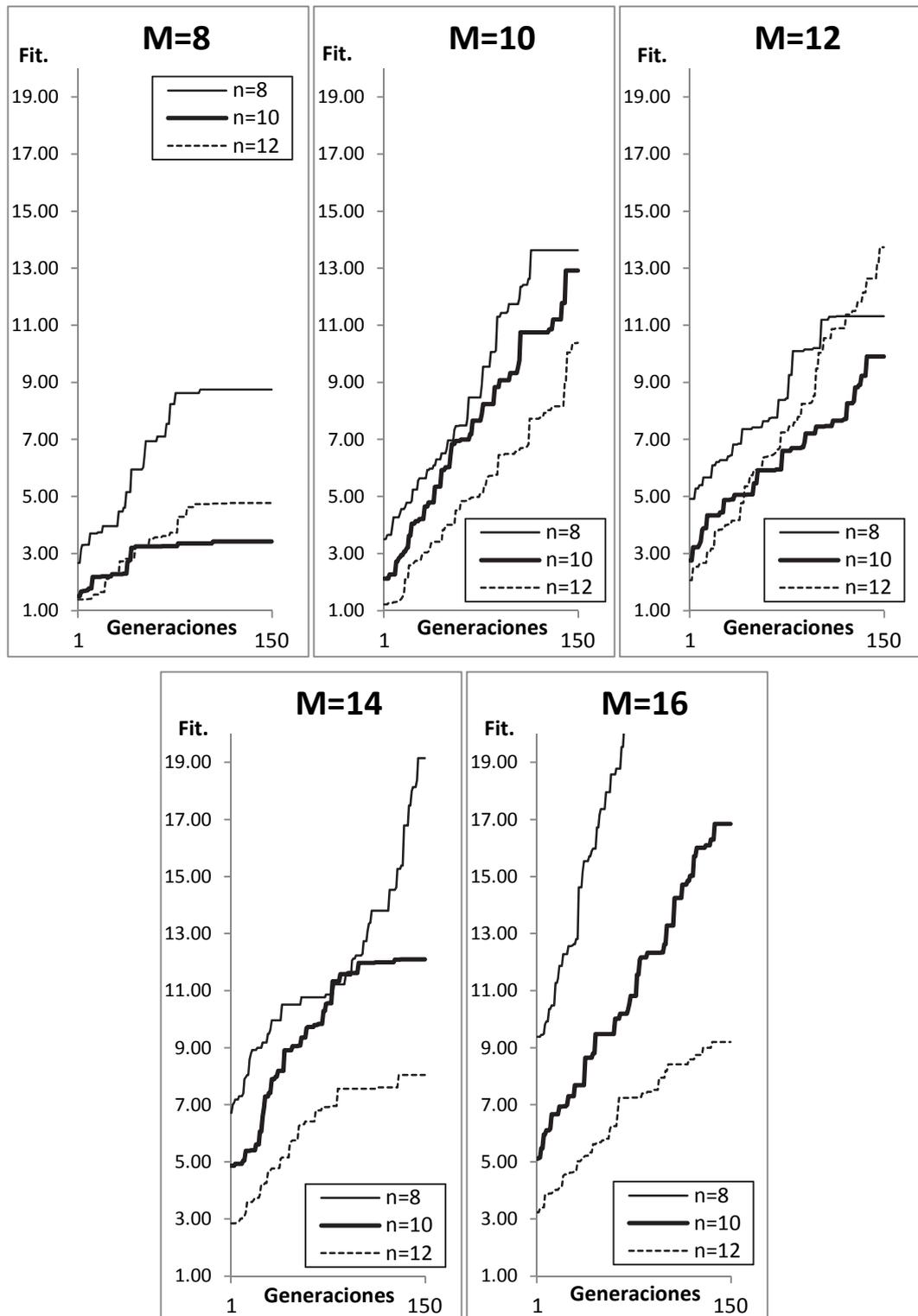


Figura 25. Evolución de la solución óptima de las 15 instancias experimentales a lo largo de 150 generaciones. La comparativa se ha hecho según el tamaño de las palabras del código corrector.

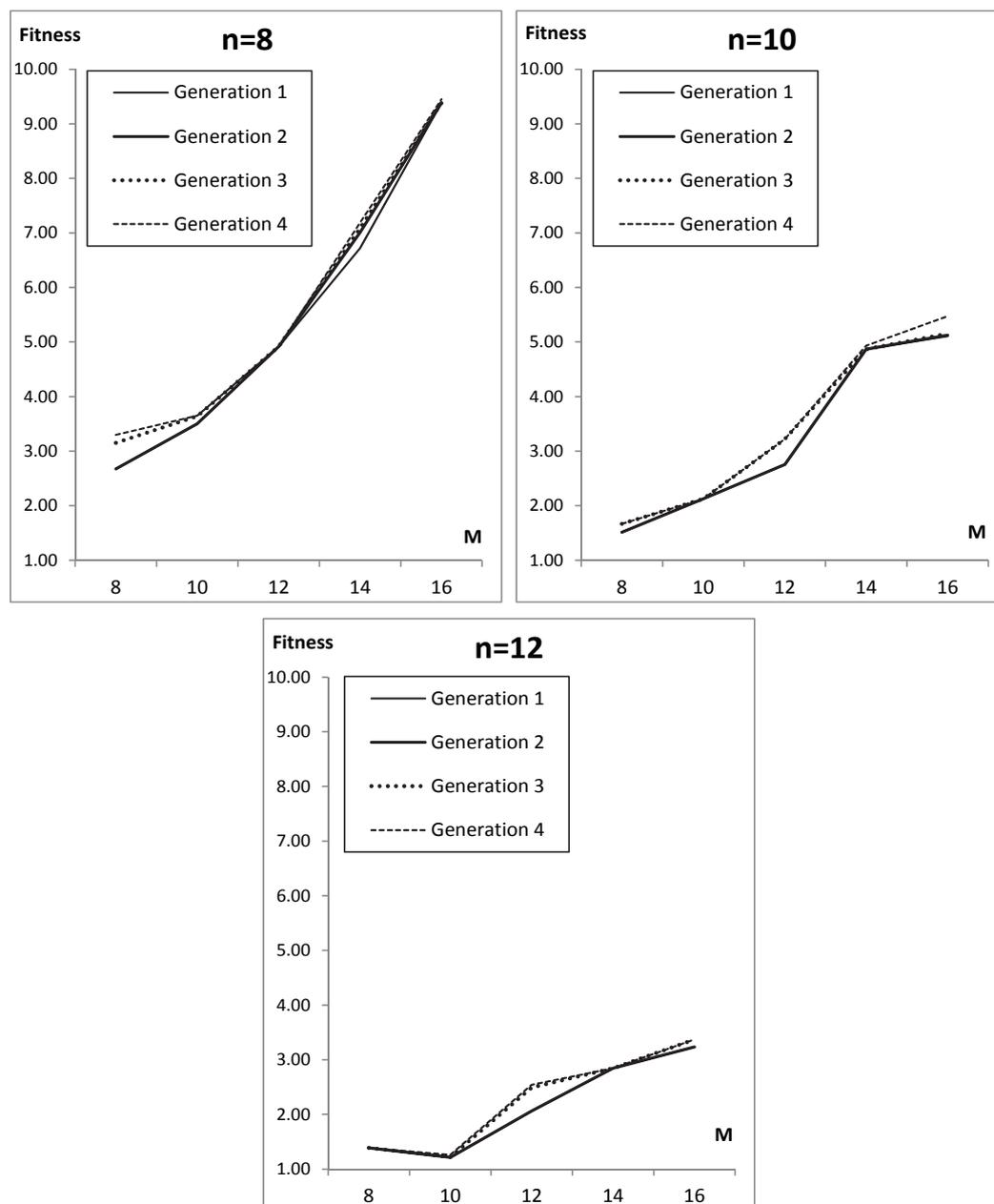


Figura 26. Valor del *fitness* de la solución óptima de las 15 instancias experimentales, para las primeras 4 generaciones del algoritmo genético.

Algoritmo	Mejor fitness	Comentarios
AG	7.86	AG implementación de esta tesis
RA	3.16	Algoritmo repulsivo
CHC	8.43	AG con control de endogamia
ssGA	7.06	AG de estado estacionario
cGA	7.75	AG celular

Tabla 7. Comparativa de resultados para la misma instancia experimental.

En la Figura 27 se muestra el ECC correspondiente al diseño óptimo encontrado al aplicar el AG, cuyo valor de *fitness* se muestra en la **Tabla 7**.

1	0	1	1	1	0	1	1	1	0	1	0
1	1	0	0	0	1	1	0	1	1	0	0
1	0	1	0	1	1	1	1	1	0	1	0
0	1	0	0	1	0	1	0	1	0	1	1
1	0	0	0	0	1	0	0	1	0	1	0
1	1	0	0	1	1	1	0	1	0	0	0
1	1	0	1	0	1	0	1	0	1	0	1
0	0	1	0	1	1	1	1	0	1	0	0
1	0	0	1	0	1	0	1	0	1	0	1
1	0	1	1	0	1	1	0	1	1	0	0
0	0	1	1	1	0	1	1	0	1	0	0
0	0	1	1	1	1	0	0	1	0	0	1
0	0	0	0	0	0	1	0	0	1	1	1
1	1	0	0	1	0	1	0	0	0	1	0
1	1	0	0	0	1	0	0	1	1	1	1
0	0	1	1	1	1	0	0	1	1	0	0
1	1	0	1	0	0	0	1	0	0	1	1
1	0	0	1	0	0	0	0	0	0	0	1
0	0	1	1	1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	1	0	1	0	1
0	0	0	1	0	1	1	1	0	1	1	1
0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	1	0	0	0	0	1	0	1	0
0	0	1	1	0	0	0	0	1	0	1	1

Figura 27. ECC obtenido por el AG para la instancia experimental considerada.

#### 4.6.7 Servicio web de apoyo a la experimentación

Como ayuda a la experimentación con el AG desarrollado, y para explorar las capacidades de los servicios web para facilitar la investigación en optimización combinatoria, se desarrolló un servicio web que ejecuta el AG con instancias de cualquier tamaño, reportando de forma gráfica y texto los resultados obtenidos.

La tecnología utilizada para desarrollar el servicio web es análoga a la que implementa otros dos servicios web que se citan en esta tesis: MOSET y TIDESI [Gómez-Pulido 2006]. Sobre un servidor con S.O. Microsoft Windows 2003 server y con tecnología CGI (*Common Gateway Interface*, tecnología precursora de *Active Server Pages*, ASP) para ejecutar código MATLAB, se creó una página web (<http://oplink.unex.es/ecc>) sobre la cual el usuario podía especificar el tamaño de la instancia experimental, el tamaño de la población, el número de generaciones y el número de ejecuciones por experimento.

El servicio web finaliza la ejecución de un experimento cualquiera mostrando los siguientes resultados de forma gráfica y en formato texto para su descarga en archivo (Figura 28) para su posterior análisis:

- Valor óptimo medio.
- Valor mínimo.
- Valor máximo.
- Desviación.
- Tiempo de computación.

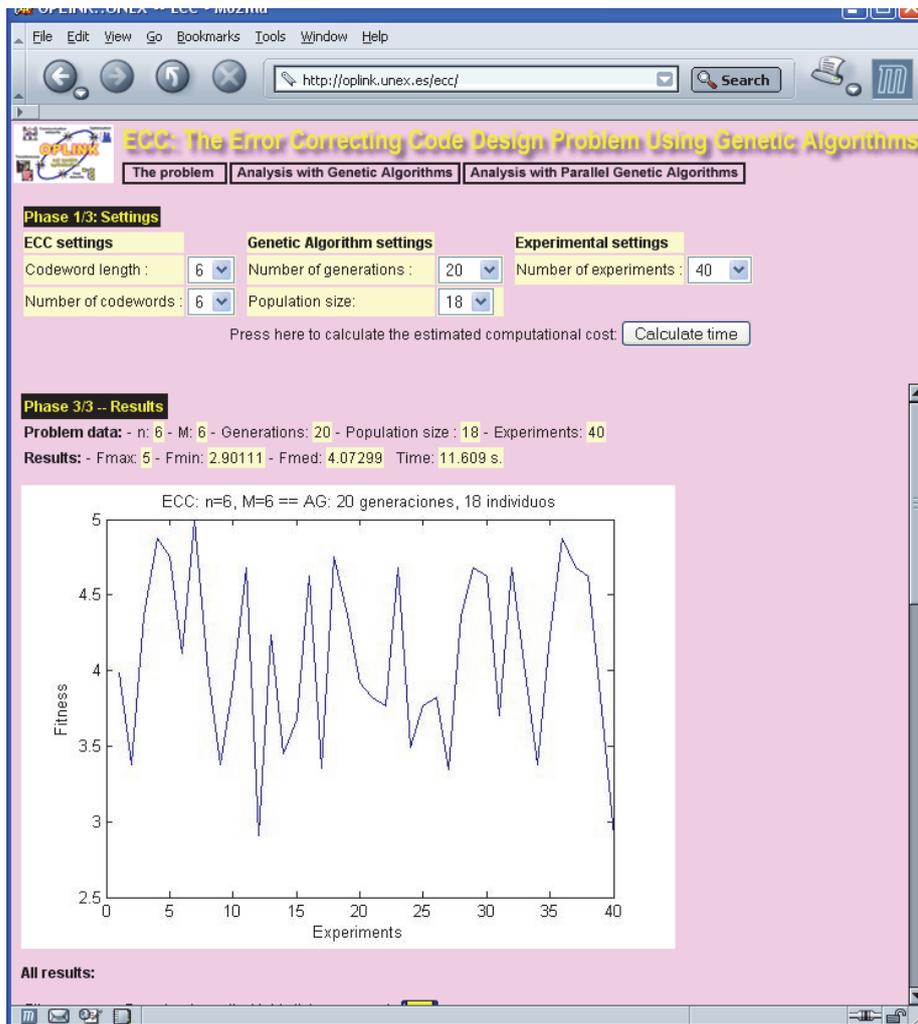


Figura 28. Servicio web para la experimentación con AG para el problema ECC.

## **4.7 Conclusiones**

Abordar este problema mediante la implementación desarrollada del AG ha sido una tarea satisfactoria en cuanto a que los resultados obtenidos han sido buenos al contrastarse con los obtenidos por otras heurísticas, y que fueron publicados en congresos científicos internacionales.

El coste computacional asociado a instancias de gran tamaño puede minorarse, entre otros factores, mediante dos formas:

- Ajustando el tamaño de población.
- Reduciendo el número de generaciones

En cualquier caso, se ha constatado experimentalmente que la metodología en sí misma, con sus parámetros bien sintonizados, produce buenos resultados para instancias de menor tamaño.

## **4.8 Aportaciones**

Como aportaciones realizadas por la doctoranda en este problema de optimización del diseño de códigos correctores de errores en comunicaciones, además del propio valor de adquisición de conocimientos y desarrollo de habilidades y experiencia en computación evolutiva, se citan las siguientes:

- Proponer, ante problemas con instancias que requieren un alto coste computacional, una metodología de experimentaciones previas con instancias más pequeñas (y por tanto, de computación más rápida), que permita realizar el ajuste de parámetros del algoritmo genético. El análisis de los resultados en el problema de optimización de códigos correctores de error puede extrapolarse a casos de instancias más grandes.
- Desarrollar una implementación de algoritmo genético con parámetros bien sintonizados que ha permitido mejorar las marcas obtenidas por las heurísticas evolutivas publicadas por otros autores.

En cuanto a la productividad científica, las anteriores aportaciones aparecen en las siguientes publicaciones:

- **Conferencias.**
  - Jaraíz-Simón, M.D., Gómez-Pulido, J.A., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M., Granado, J.M. "A Genetic Algorithm to Design Error Correcting Codes", in: 13th IEEE Mediterranean Electrotechnical Conference, IEEE, Benalmádena, Málaga, Spain, 2006, 807-810.
  - Gómez-Pulido, J.A., Jaraíz-Simón, M.D., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M. "Procesador paralelo para la medida de la calidad del diseño de códigos correctores de errores en comunicaciones", in: Actas de las VIII Jornadas sobre Computación Reconfigurable y Aplicaciones (JCRA 2008), Univ. Rey Juan Carlos, Madrid, España, 2008, 315-320.

# ***5 Optimización de Redes Móviles Ad- Hoc***

*“Reunirse es un comienzo.  
Permanecer juntos es un progreso.  
Trabajar juntos es el éxito.”*

**Henry Ford.**  
Empresario estadounidense (1863-1947).

## 5.1 Resumen

Las redes **MANET** (*Mobile Ad-Hoc NETWORKS*) son redes de comunicaciones inalámbricas que se forman cuando un número de terminales móviles se auto-organizan de forma rápida sin infraestructura previa para la comunicación. La movilidad y el alcance limitado de las conexiones inalámbricas hacen impredecible la topología de la red, que es cambiante en el tiempo.

En este trabajo de tesis se resuelve un problema de optimización multi-objetivo que aparece al considerar la difusión (broadcasting) en este tipo de redes, cuando un terminal envía un mensaje a todos los demás. Una difusión eficiente y fiable es un reto en MANET, especialmente en escenarios metropolitanos, caracterizados por una alta densidad de nodos heterogéneos y de gran movilidad.

Para el modelado de la red MANET se ha utilizado el simulador *Madhoc*, que puede implementar diferentes escenarios (áreas abiertas, entornos metropolitanos, autopistas, etc.). Para este trabajo se ha utilizado un escenario de centro comercial.

Se ha considerado el protocolo de difusión de Inundación Retardada con Vecindad Acumulativa ó *Delayed Flooding with Cumulative Neighborhood (DFCN)*, para el cual se han ajustado sus parámetros de forma que se satisfagan tres objetivos simultáneos y contrapuestos entre sí:

- Maximizar el número de dispositivos alcanzados (cobertura).
- Minimizar la utilización de la red (ancho de banda).
- Minimizar la duración del proceso.

Los parámetros que configuran y determinan el comportamiento de DFCN son cinco: mínimo valor para retransmisión, retraso de valoración aleatorio (mínimo y máximo), densidad de la red y umbral de densidad segura.

Para la resolución de este problema de optimización multi-objetivo se han utilizado los algoritmos NSGA-II y Evolución Diferencial Multi-objetivo, encontrando el mejor comportamiento para el primero. No obstante, ambos algoritmos muestran frentes de soluciones similares, por lo que se optó por aumentar el número de evaluaciones para evidenciar mejor su diferencia.

De esta forma, y para afrontar el problema de afrontar unos tiempos de computación muy elevados, se diseñó una estrategia de computación paralela basada en el modelo de islas, donde cada población se divide en un número de sub-poblaciones independientes. En este caso se confirmó el mejor comportamiento de NSGA-II, pero alcanzando precisiones y rendimientos mayores para las soluciones óptimas (utilizando la métrica hipervolumen y tests estadísticos), y reduciendo el tiempo de computación necesario.

## 5.2 Introducción

En los últimos años se ha producido una gran proliferación de dispositivos móviles, como portátiles, PDAs, *tablets* o *smartphones*, al incrementarse su potencia de cálculo y capacidades de comunicación inalámbrica. Además, este crecimiento se debe también a la movilidad y autonomía que ofrecen al usuario, pudiendo ser utilizados sin tener que estar obligatoriamente en un lugar fijo.

Con el auge del uso de Internet en la llamada "Sociedad de la Información", se demuestra que una de las principales motivaciones de las personas es la de conocer e intercambiar información con otras. Por ello, hoy en día es imprescindible disponer de una tecnología que permita que varios dispositivos móviles puedan comunicarse entre sí, como por ejemplo puede ser Bluetooth, Wi-Fi, WiMax o 3G, entre otras. Estas tecnologías inalámbricas proporcionan una libertad e independencia que nunca podrán ser igualadas por las conexiones de cable, aunque también es cierto que son algo más limitadas en otros aspectos (duración de la batería, mayor complejidad, menor ancho de banda, mayores amenazas en la seguridad, etc.).

Son muchos los protocolos de comunicación y tecnologías relacionadas que se han propuesto para las redes móviles **ad-hoc**. Algunos de estos protocolos se diseñaron específicamente para las redes ad-hoc y otros se adaptaron de soluciones existentes para otras redes. Los **protocolos de encaminamiento** permiten, desde un nodo origen, hacer llegar su mensaje a un nodo destino, usando para ello la mejor ruta posible, y confiando en la colaboración del resto de nodos para su transporte mediante retransmisión.

Una red ad hoc consiste en un grupo de ordenadores o dispositivos con capacidad de procesamiento y comunicación que se enlazan entre ellos directamente a través de señales de radio sin puntos de acceso. Las configuraciones ad hoc son comunicaciones de tipo punto a punto **P2P** ("*peer to peer*"), que en el caso de las MANET solamente los dispositivos dentro de un rango de transmisión definido pueden comunicarse entre ellos.

En principio, el modo ad-hoc sólo permite la transmisión de paquetes a los dispositivos que están en su área de cobertura, pero las aplicaciones requieren conectarse con otros nodos que están situados a una distancia mayor. Para ello, se pueden utilizar los dispositivos vecinos como *routers* para que reenvíen la información al destinatario, como es el caso de las MANET; a esto se le denomina tecnología **multisalto** (pues la información da varios saltos desde que sale del origen hasta que llega al destino).

Por ejemplo: un portátil quiere enviar un mensaje a un teléfono móvil, que está fuera de su área de cobertura. Si la red soporta la opción multisalto, el portátil podría intercambiar información con el teléfono móvil a través de una PDA (nodo intermediario), que en este caso actuará como router reenviando la información. Así, para que una MANET tenga capacidad multisalto, se necesita un protocolo de encaminamiento que establezca la topología adecuada de red, definiendo los dispositivos que actuarán como routers y las rutas de tráfico para saber a qué nodo se deben enviar los paquetes.

El trabajo de optimización abordado en esta tesis se centra en las redes MANET, que se caracterizan por no requerir infraestructuras previas ni necesitar de un control centralizado (que no existe ningún nodo central o punto de acceso fijo), donde todos los nodos que participan en la red son considerados iguales. En estas redes, el número de nodos es variable, así como la velocidad a la que se desplazan, configurando una topología totalmente dinámica.

Al tratarse de nodos móviles, estos cuentan con sistemas de comunicaciones inalámbricas (con restricciones de ancho de banda) y una batería de duración limitada. Debido a estas restricciones, la mayoría de protocolos y aplicaciones utilizados habitualmente deben ser modificados y optimizados.

### **5.2.1 Redes inalámbricas Ad Hoc**

Las redes ad hoc surgieron de una necesidad militar que pretendía conectar distintas unidades tácticas sin requerir la presencia de una red fija. Posteriormente se produjo una diversificación en el acceso a Internet, destacando las redes inalámbricas con o sin infraestructura fija.

Si bien las más conocidas son las que cuentan con una infraestructura fija, hay casos donde no es posible disponerla. Es en este punto donde cobran gran importancia las redes Ad Hoc, en las que los terminales actúan no sólo como receptores o emisores sino también como encaminadores.

#### **5.2.1.1 Definición**

Una red Ad hoc es aquella en la que no hay un nodo central, sino que todos los dispositivos están en igualdad de condiciones. Estas redes se crean espontáneamente, sin una infraestructura específica y funcionan en un espacio y tiempo limitados. Proporcionan flexibilidad y autonomía mediante principios de auto-organización. Por tanto, las redes ad hoc consisten en nodos que se comunican inalámbricamente y que no requieren de una infraestructura preestablecida [Bangnan 2003]. Esto implica que no tienen control centralizado y son flexibles y fácilmente desplegados.

Hay dos perspectivas para configurar la red. La primera considera que cada uno de los nodos está dentro del rango de cobertura inalámbrica de los otros; de esta forma, la red ad hoc está completamente conectada. Sin embargo, una conectividad total puede ser inviable en la práctica, no sólo por la dificultad para lograr una cobertura plena, sino por el desperdicio de potencia para los terminales, mayoritariamente operados por baterías.

La segunda perspectiva considera la distribución de nodos en la red sin estar necesariamente en el área de cobertura de otros nodos inalámbricos; ante la ausencia de una infraestructura de enrutadores disponible, es necesario un enrutamiento multisalto entre los nodos. Los dispositivos pueden descubrir a otros cercanos para formar la red. Pero, también, los dispositivos pueden buscar nodos fuera del área de alcance a través de enrutamiento multisalto, para conectarse con otros dispositivos ya conectados a la red y a su alcance mediante múltiples nodos.

Esta característica evoluciona el concepto de redes ad hoc hacia el concepto de redes Mesh o redes de infraestructura instantánea [Bruno 2005].

Las redes ad hoc presentan cambios de topología debido a la movilidad, surgiendo así nuevos paradigmas en la evolución de esta tecnología para ofrecer soluciones en problemas tales como topología dinámica y recursos de ancho de banda.

La expresión ad hoc está contenida en el estándar IEEE 802.11 [Anastasi 2003], donde se contempla la configuración de operación ad hoc de las estaciones en la capa **MAC** (Control de Acceso al Medio), de forma que sea posible la comunicación directa entre ellas. Este estándar ha impulsado el crecimiento de las redes móviles. Por otro lado, el auge de los dispositivos portátiles y móviles en los años 90 empujó a la IETF (*Internet Engineering Task Force*) a crear el grupo de trabajo MANET (*Mobile Ad hoc Networking group*) para estandarizar las redes ad hoc y estimular su investigación. Estos desarrollos se enfrentan a grandes retos tecnológicos: problemas de optimización del ancho de banda disponible, acceso múltiple, control de potencia, capacidad variable de canal y seguridad; técnicas para mantenimiento y configuración de topologías de red, problemas de encaminamiento atendiendo a esquemas multisalto [Murray 2010], etc.

#### **5.2.1.2 Clasificación**

Hay varias opciones para establecer una clasificación de estas redes. Por ejemplo, atendiendo a la infraestructura de red:

- Redes ad hoc con infraestructura: los nodos se mueven mientras se comunican con una estación base fija. Cuando un nodo se mueve fuera del rango de una estación fija entra en el alcance de otra estación.
- Redes ad hoc sin infraestructura: no hay estaciones base fijas y todos los nodos de la red actúan como routers. Las MANET son redes sin infraestructura.

Otra clasificación consiste en redes de un solo salto o multisalto.

- Los nodos de las redes de un solo salto se comunican solo con los nodos que tiene a su alcance.
- En las redes multisalto los nodos que no pueden comunicarse directamente utilizan otros para retransmitir la información. Las MANET son multisalto.

Por último, una clasificación que ve las redes MANET como un tipo independiente incluye:

- Redes móviles ad hoc (MANET).
- Redes de sensores inalámbricos.
- Redes híbridas.

A su vez, las redes MANET son de dos tipos según su conectividad a otras redes:

- MANET autónomas (no están conectadas a ninguna red y los nodos se identifican por una IP distinta).
- MANET subordinadas (conectadas a una o más redes externas).

## **5.2.2 MANET**

Las redes inalámbricas móviles MANET son objeto de muchas líneas de investigación con el objetivo de mejorar su funcionamiento. Los estudios se centran en campos como calidad de servicio, seguridad, autonomía, etc.

Las redes MANET se apoyan en tecnologías de la comunicación inalámbrica tales como 3G HSPA, WLAN (estándar 802.11), Bluetooth, WiMAX o LTE. La mayoría de estas tecnologías inalámbricas se basan en el principio de una comunicación punto a punto, donde los nodos móviles se comunican directamente con un punto de acceso centralizado (que suele ser fijo y conectado a otras redes a través de cable, usualmente Internet). Pero una red MANET se basa en una red descentralizada y multisalto, donde los nodos intermedios actúan como encaminadores o *routers*, reenviando la información en cada salto.

### **5.2.2.1 Definición**

Una MANET puede ser definida como una red auto-configurable de enrutadores móviles y sus hosts asociados, conectados a través de enlaces inalámbricos formando una topología arbitraria, donde los nodos pueden moverse libremente y organizarse aleatoriamente; así, una red MANET puede operar de forma autónoma o estar conectada a una red más grande. Se pueden considerar a las redes MANET como especialización de las redes Mesh para entornos móviles [Akyildiz 2005].

La movilidad de los nodos hace que sus enlaces estén continuamente rompiéndose o estableciéndose, con incorporaciones o abandonos continuos a la red, por lo que la topología de la red cambia mucho. Todos los nodos pueden actuar tanto como emisores, receptores y routers, ya que las rutas para llegar a un destino pueden tener varios saltos.

La necesidad de crear redes de forma rápida en lugares sin infraestructura suele implicar que los nodos exploren el área y, en algunos casos, se deban unir para conseguir un objetivo. El tipo de movilidad que desarrollen los nodos puede tener una influencia a la hora de escoger el protocolo de encaminamiento que aumente el rendimiento de la red.

### **5.2.2.2 Características**

Las redes MANET poseen características que las hacen atractivas para entornos que requieren una alta movilidad y flexibilidad en la configuración de la red. En primera instancia, la capacidad multisalto es esencial para que las redes MANET logren dos objetivos importantes: por un lado, extender la cobertura sin sacrificar la capacidad del canal y por otro, proporcionar alta conectividad. Como consecuencia, se logran tasas de transferencia más altas a través de enlaces cortos sin sacrificar cobertura, con menos interferencia entre los nodos y una reutilización de frecuencia más eficiente. Estas características imponen retos importantes en los protocolos de enrutamiento, así como también en la configuración y despliegue de la red.

Al carecer de infraestructura fija previa, las redes MANET adolecen de seguridad o control centralizados, de forma que para que la red sea funcional se debe asumir que los nodos que la forman son capaces de seguir un protocolo que los auto-organice. Igualmente, y para que los nodos puedan ver cubiertas sus necesidades de comunicación, se requiere que todos los nodos intermedios entre el emisor y el receptor de un mensaje, y que forman la mejor ruta emisor-receptor, reenvíen dicho mensaje al siguiente salto de esa ruta.

La idea principal de la llamada auto-organización se basa en la coordinación y colaboración de todos los nodos de la red para conseguir un mismo objetivo. La autoconfiguración puede desglosarse en las siguientes capacidades:

- Auto-reparación: mecanismos que permitan detectar, localizar y reparar automáticamente los fallos siendo capaces de distinguir la causa del error.
- Auto-configuración: métodos de generación de configuraciones adecuadas en función de la situación actual dependiendo de las circunstancias ambientales.
- Auto-gestión: capacidad de mantener dispositivos o redes dependiendo de los parámetros actuales del sistema.
- Adaptación: adecuación a los cambios de las condiciones ambientales.

Por otro lado, el dinamismo de las MANET permite pasar por varios estados de operación que se clasifican de acuerdo con la tasa de conexiones y desconexiones. Así, una MANET es:

- Estable: maneja una tasa baja de conexiones y desconexiones.
- De alta conexión: existe una tasa alta de conexiones y baja de desconexiones.
- De alta desconexión: la tasa de desconexiones es alta y la de conexiones, baja.
- Altamente dinámica: existe una tasa elevada de conexiones y desconexiones.

Otras características de las MANET son: ausencia de infraestructura, terminales autónomos, conexiones inalámbricas, funcionamiento distribuido, topología dinámica, ancho de banda limitado, variación en la capacidad de los enlaces y los nodos, limitaciones en el consumo y conservación de energía, escalabilidad, baja seguridad, encaminamiento multisalto, entorno imprevisible, etc.

### **5.2.2.3 Ventajas e inconvenientes**

Las ventajas surgen directamente de su forma de despliegue:

- No dependen de ninguna infraestructura concreta preexistente, por lo que se pueden desplegar en cualquier lugar en que sea necesario, en interior o exterior.
- Son baratas y fáciles de desplegar. En las redes comunitarias, el despliegue no tiene coste (salvo el precio del terminal), y con el uso de los protocolos adecuados su despliegue consiste solamente en encender los nodos y dejar que se auto-organicen.

- Toleran cierta cantidad de fallos, dependiendo del número de nodos y de rutas posibles entre cada par origen-destino, de la velocidad de cada nodo y de las trayectorias que siguen, readaptando las rutas cuando sea conveniente en función de la topología de la red en cada momento. En la Figura 29 se muestra un ejemplo de cómo cambia la topología de red, en la que la ruta entre A y D, que pasaba por C a dos saltos, ahora debe pasar también por E a tres saltos.

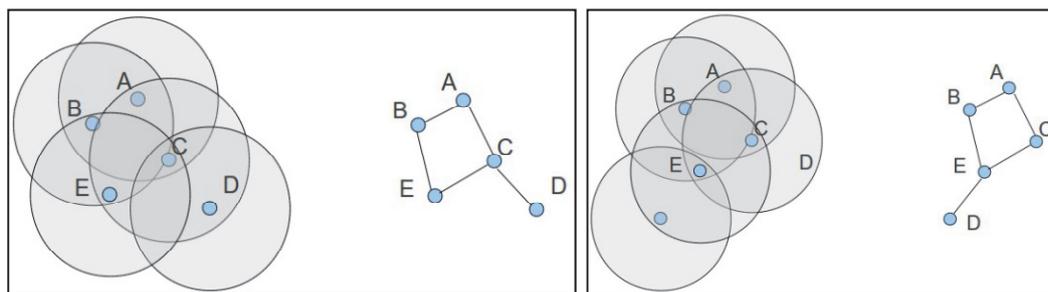


Figura 29. Cambio de topología de MANET de 5 nodos, donde se mueve el nodo D.

Sin embargo, las MANET adolecen de una serie de inconvenientes o debilidades que hay que tener en cuenta:

- Todos los nodos que forman la red deben ejecutar los mismos protocolos para entrar a formar parte o salir de la misma, buscar las mejores rutas para los destinos de paquetes, notificar su presencia a los vecinos y detectar su presencia, etc., pues, de otro modo, no existiría la MANET, o habría tantas como protocolos estuvieran en ejecución.
- Como red inalámbrica, sufre problemas a nivel físico y de acceso al medio: alcance reducido, pérdida de paquetes por errores de transmisión, etc. La red también sufre el conocido “problema del nodo oculto”, que se da cuando las transmisiones de dos nodos que no se conocen colisionan en un nodo que es vecino de ambos a la vez.
- Son vulnerables por utilizar el medio aéreo, por lo que se tendrían que implementar mecanismos de seguridad de las comunicaciones.
- La productividad y existencia de la red depende de las velocidades relativas de los nodos. Si éstos se mueven muy rápido, pronto quedan fuera del alcance de otros nodos, particionando la red o aislándose.

#### 5.2.2.4 **Clasificación**

Las MANET pueden ser subordinadas o autónomas en función de si están conectadas o no a una red externa. En una MANET autónoma los nodos pueden identificarse unívocamente a través de una dirección IP con la única premisa de que ésta sea distinta a la de cualquier otro nodo de la red. Sin embargo, en una MANET subordinada se obliga a usar un direccionamiento IP topológico correcto y enrutable globalmente.

### **5.2.2.5 Aplicaciones**

Al no tener una infraestructura fija, las redes MANET son fácilmente desplegables, por lo que son muy útiles en entornos donde resulte costoso instalar una infraestructura fija, donde las características físicas del entorno no lo permitan o donde se requiera un despliegue rápido. Algunas de las aplicaciones que pueden tener las MANET son: entornos militares, situaciones de emergencia, entornos civiles y redes de área personal.

### **5.2.2.6 VANET**

Una red ad hoc vehicular, conocida como **VANET** (*Vehicular ad hoc NETWORK*) es un tipo de red MANET donde los nodos son vehículos o la propia infraestructura del tráfico, como por ejemplo las señales verticales. Estas redes son utilizadas para comunicarse entre vehículos y/o entre vehículos e infraestructura al borde de la carretera (estaciones base) para compartir información de tráfico, estado de las carreteras, condiciones medioambientales y posibles accidentes-obras.

Al ser una extensión de las MANET, las redes VANET presentan los mismos retos, incluso si cabe más complicados, debido a la velocidad a la que se mueven los vehículos, lo que afecta a los protocolos de comunicación y las soluciones de seguridad, entre otros factores.

## **5.2.3 *Funcionamiento de las redes MANET***

En esta sección se expone el funcionamiento de las redes MANET y de los principales protocolos que se utilizan en los distintos niveles, información que es muy útil para comprender la naturaleza del problema de optimización que se aborda en esta tesis doctoral.

### **5.2.3.1 Capa física**

En las redes MANET los nodos se comunican usando como medio de transmisión el espacio libre, a través de canales de radiofrecuencia (RF) o de rayos infrarrojos (IR), si bien este último medio es poco habitual debido a sus desventajas.

Los estándares IEEE 802.11X definen interfaces para los canales de radiofrecuencia en las bandas de 2,4GHz y de 5GHz. La banda de 2,4GHz es de uso muy abierto y está más saturada, ya que se utiliza para otro tipo de redes, tales como Bluetooth, microondas, radio control, etc. Esto provoca que se hagan presentes más interferencias que en las bandas de 5GHz; sin embargo, sus señales sufren muchas menos pérdidas de transmisión en el espacio libre.

Los canales RF utilizan antenas omnidireccionales: son económicas y permiten que los dispositivos móviles se comuniquen con cualquier otro nodo que se encuentre lo suficientemente cerca en cualquier dirección. La desventaja es que dos nodos distintos pueden transmitir al mismo tiempo usando el mismo canal de RF, por lo que hace falta un protocolo MAC (Medium Access Control) que se encargue de evitarlo.

### 5.2.3.2 Capa MAC

Existen dos familias principales de protocolos MAC: de acceso aleatorio y de acceso controlado. En los primeros los nodos compiten entre ellos para conseguir el canal de transmisión, mientras que en segundos existe un dispositivo que decide cuál de los nodos puede acceder al medio en cada momento.

Hay muchos protocolos MAC de acceso controlado, entre los cuales destacan: FDMA (*Frequency Division Multiple Access*), TDMA (*Time Division Multiple Access*) y CDMA (*Code Division Multiple Access*). Estos protocolos se aplican principalmente en las redes que cuentan con una infraestructura y con un nodo maestro que se encarga de administrar el acceso al medio de transmisión por parte de los nodos de la red.

Las MANET, en cambio, usan protocolos MAC de acceso aleatorio, concretamente el CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). El estándar IEEE 802.11 define el uso de los dos niveles más bajos de la arquitectura OSI (*Open Systems Interconnection*), que son la capa física y la capa de enlace, y se ha convertido en uno de los estándares más populares.

Para mejorar el comportamiento de los protocolos MAC para las MANET hay varias propuestas algorítmicas que tratan el consumo de energía, la gestión de paquetes, la reducción de los radios de transmisión y de recepción, los diferentes esquemas de codificación y modulación de la señal, etc.

### 5.2.3.3 Capa de red

Es en esta capa dónde más se caracterizan las redes MANET de otras. Los protocolos de encaminamiento para MANET han de adaptarse rápidamente a los cambios constantes de la topología de la red para poder mantener una ruta para la comunicación entre los nodos. Existe una gran cantidad de protocolos de encaminamiento para las redes ad hoc, los cuales pueden ser más o menos adecuados en cada escenario en concreto.

El protocolo de encaminamiento establece y mantiene las rutas por donde viaja toda la información que genera la red, es decir, que cada dispositivo sepa a qué otro nodo de la red dentro de su rango de transmisión debe enviar cada paquete. Para ello, cada dispositivo posee una tabla de *routing* donde aparece dicha información, siendo la misión del protocolo la de crear y mantener esas tablas correctamente, definiendo así la topología de la red.

Estos protocolos especifican la estructura de los mensajes que se intercambian los dispositivos para que conozcan la topología de la red y de esta manera actualizar la tabla de *routing*. Ejemplos típicos de protocolos de *routing* para redes IP fijas son RIP (*Routing Information Protocol*) y OSPF (*Open Shortest Path First*).

Al contrario que en las redes fijas (o móviles en modo infraestructura), todos los dispositivos pueden actuar como *routers*, por lo que cada emisor tiene una ruta específica por la que enviar los paquetes a cada destino de la red MANET. Por esta razón los protocolos de *routing* habituales en redes fijas no son apropiados para MANET. Además, hay que tener en cuenta otras propiedades que hacen el diseño del protocolo de *routing* bastante más complejo, ya que la comunicación se

produce de forma distribuida (y no de forma centralizada vía *routers* como en las redes fijas), la topología cambia con mucha rapidez y el ahorro de energía es fundamental.

Las principales características de un protocolo de encaminamiento MANET son:

- Creación y organización de rutas de tráfico de forma automática y autónoma.
- Capacidad de multi-salto, donde las rutas no presenten bucles infinitos.
- Mantenimiento de la topología de red hasta en las redes más dinámicas.
- Convergencia rápida de la red.
- Mínima sobrecarga en el tráfico de la red.
- Que se pueda utilizar en grandes redes con muchos dispositivos (escalabilidad).

No hay ningún protocolo que responda de forma satisfactoria a todas estas características, por lo que la elección del protocolo de encaminamiento dependerá del número de nodos, de su movilidad y de la cantidad de tráfico de red.

Podemos clasificar los protocolos para MANET según varios aspectos:

- Dependiendo del mecanismo de actualización de las mejores rutas origen-destino, un protocolo puede ser reactivo (cuando busca la mejor ruta en el momento necesario para enviar un mensaje) o proactivo (si está continuamente actualizando su tabla de rutas para asegurar proporcionar instantáneamente la mejor ruta para el mensaje).
- Según dónde se toma la decisión de encaminamiento de cada paquete, un protocolo puede basarse en encaminamiento fuente (si la decisión se toma en el nodo origen para toda la ruta del mensaje, indicando explícitamente por cuáles nodos ha de pasar) o distribuido (si la decisión de por cuál enlace ha de encaminarse el paquete se toma en cada nodo de la ruta origen-destino).
- Dependiendo de las posibilidades de cambiar de ruta una vez inyectado el paquete en la red, un protocolo puede ser adaptativo, si permite cambiar de ruta en respuesta a fallos de enlaces o congestión; o determinista, si no lo permite.
- Dependiendo del parámetro que se use para decidir la mejor ruta, hay protocolos para el camino más corto, la ruta de menor latencia, etc.
- Dependiendo de si el protocolo utiliza o no información topológica, hay protocolos de encaminamiento plano o de encaminamiento jerárquico.

Además, hay protocolos de encaminamiento MANET basados en el nivel de las baterías, para equilibrar el consumo medio en toda la red.

Los protocolos tienen en cuenta siempre el descubrimiento de rutas (usando paquetes de control) y la transferencia de mensajes. Los protocolos, en función cómo determinan las rutas, son reactivos, proactivos e híbridos. Además, existen protocolos que también consideran Calidad de Servicio (**QoS**) y anonimato en las comunicaciones.

- Los protocolos reactivos o bajo demanda hallan las rutas entre un origen y un destino bajo demanda de la fuente (sólo cuando sea necesario iniciar una transmisión se busca una ruta). Una vez encontrada la ruta, los nodos participantes se encargarán de mantener la transmisión.
- Los protocolos proactivos mantienen tablas de rutas actualizadas, por lo que cada nodo ha de actualizar una tabla con todas las rutas hacia los otros. La información de estas tablas se actualiza periódicamente y ante cualquier cambio de la topología.
- Los protocolos híbridos combinan características de los anteriores, agrupando los nodos para que, cuando necesiten rutas hacia uno cercano, usen un encaminamiento proactivo, y para descubrir rutas en lejanos usen uno reactivo.
- Los protocolos QoS buscan rutas que satisfagan los requisitos de QoS establecidos, cumpliendo restricciones del retardo, ancho de banda, etc.

#### **5.2.3.4 Capa de transporte**

En las redes MANET se busca la máxima compatibilidad con los protocolos de Internet, ya que el objetivo final es poder ofrecer Internet en todo lugar y momento. Por esta razón, MANET se basa en la pila de protocolos TCP/IP.

Existen algunas propuestas con el fin de mejorar las prestaciones de los principales protocolos de transporte de Internet, TCP y UDP, en entornos Ad Hoc. Por otro lado, se han desarrollado protocolos de transporte para otras aplicaciones de tiempo real y *streaming*, que establecen múltiples flujos de datos entre dos nodos para una misma comunicación y funcionan junto a protocolos de encaminamiento multi-ruta.

#### **5.2.3.5 Capa de aplicación**

Las aplicaciones diseñadas para Internet se pueden soportar también en las redes MANET. En general, las aplicaciones de transmisión de datos que no precisen de grandes anchos de banda ni altas limitaciones en cuanto al tiempo de entrega de los paquetes, se pueden soportar con la filosofía *Best Effort*.

No obstante, existe hoy en día en Internet una gran cantidad de aplicaciones, sobretudo multimedia, mucho más sensibles al retardo, que demandan más ancho de banda y otras garantías. Las limitaciones y características intrínsecas de las MANET dificultan ofrecer aplicaciones multimedia. Para conseguirlo, es necesario que todos los niveles de red trabajen en el mismo sentido para lograr el objetivo de poder garantizar una cierta QoS.

Se han presentado algunas propuestas a nivel de aplicación para adaptarse a las redes MANET. Estas propuestas pasan por disminuir el ancho de banda con nuevas técnicas de codificación que sacrifiquen calidad para aumentar la compresión. También hay propuestas orientadas a modificar la calidad de la información según el estado de la red.

### **5.3 Problema de difusión en MANET metropolitanas**

Como ya se ha comentado anteriormente, las redes móviles ad hoc (MANETs) están compuestas por un conjunto de dispositivos móviles de comunicación que pueden moverse libremente y organizarse entre ellos de forma totalmente arbitraria. Estas redes se caracterizan por no precisar de infraestructura previa y por ser desplegadas de forma rápida con un bajo coste. Esto implica que los dispositivos se muevan mientras se comunican y que la comunicación se establezca dentro de un rango limitado.

La movilidad de estos dispositivos y el alcance limitado de las conexiones inalámbricas hacen que la topología de la red cambie rápida e impredeciblemente a lo largo del tiempo. Por esta razón, el comportamiento dinámico de este tipo de redes supone un obstáculo para alcanzar comunicaciones eficientes.

En esta tesis se aborda el problema de la **difusión** (o *broadcasting*) en MANETs, es decir, el proceso por el cual un nodo fuente (emisor u origen) envía un mensaje a todos los nodos (receptores o destinos) en la red. El *broadcasting* es una operación común a nivel de aplicación, y es ampliamente usado para resolver muchos problemas de la capa de red, como el mecanismo básico para diversos protocolos de enrutamiento (*routing protocols*). En una MANET, y debido a la movilidad de los dispositivos, se espera que el *broadcasting* se realice con frecuencia (por ejemplo, para enviar una señal de alarma o buscar una ruta para un nodo destino). Por tanto, si se selecciona una estrategia de difusión adecuada, se contribuirá a un mayor rendimiento de la red. En resumen, el *broadcasting* es difícil de realizar de forma eficiente y fiable en el contexto de MANETs.

Concretamente, este trabajo se centra en el problema del *broadcasting* en una subclase particular de MANETs denominadas MANETs Metropolitanas [Alba 2005]. Este tipo de redes se caracterizan por su densidad, la cual es heterogénea y dinámica y porque las regiones con alta densidad no se mantienen activas todo el tiempo. Ejemplos de estas redes se pueden encontrar en aeropuertos, estaciones de tren o edificios de oficinas.

Para la estrategia de difusión se ha considerado el protocolo denominado **Inundación Retardada con Vecindad Acumulativa**. El objetivo en esta tesis no es proporcionar una estrategia genérica que funcione bien en la mayoría de las situaciones, sino conseguir ajustar de la mejor forma posible una estrategia de difusión dada, para una red concreta (MANET metropolitana). Optimizar una estrategia de difusión implica satisfacer múltiples objetivos (contrapuestos entre sí) de forma simultánea, tales como:

- Maximizar el número de dispositivos alcanzados (cobertura).
- Minimizar la utilización de la red (ancho de banda).
- Minimizar la duración del proceso.

Se trata, por tanto, de un problema multi-objetivo.

Las instancias definidas en este trabajo consideran un escenario real y complejo, como es un centro comercial (*mall*).

### 5.3.1 Redes móviles ad hoc metropolitanas

Las redes móviles ad hoc metropolitanas son MANETs con algunas propiedades particulares. En primer lugar, tienen una o más áreas donde la densidad de los nodos es mayor que la media. Se denominan áreas de alta densidad y se pueden detectar estadísticamente. Un área de alta densidad puede ser, por ejemplo, un supermercado, un aeropuerto, o una oficina.

En segundo lugar, las áreas de alta densidad no se mantienen activas todo el tiempo, es decir, pueden aparecer y desaparecer de la red. Por ejemplo, un centro comercial tiene un horario de apertura que va de 10 a 22, en horario ininterrumpido y fuera de esta franja horaria, la densidad de los nodos dentro de esta área se aproxima al valor cero.

En el contexto de MANETs metropolitanas, se pueden obtener diversas configuraciones en cuanto a las topologías. Algunos ejemplos son redes construidas de improviso por la gente moviéndose en conciertos, mercados, estaciones de tren, centros comerciales o las calles de una ciudad. Todos estos escenarios normalmente tienen características diferentes, tales como la movilidad y la densidad de dispositivos, tamaño del área y presencia o no de paredes (obstáculos para la movilidad que atenúan la intensidad de la señal), entre otras.

En este trabajo se ha utilizado **Madhoc** [Hogie 2007], un simulador de MANETs metropolitanas (Figura 30). Madhoc puede implementar diferentes escenarios (áreas abiertas, entornos metropolitanos, centros comerciales, etc.).

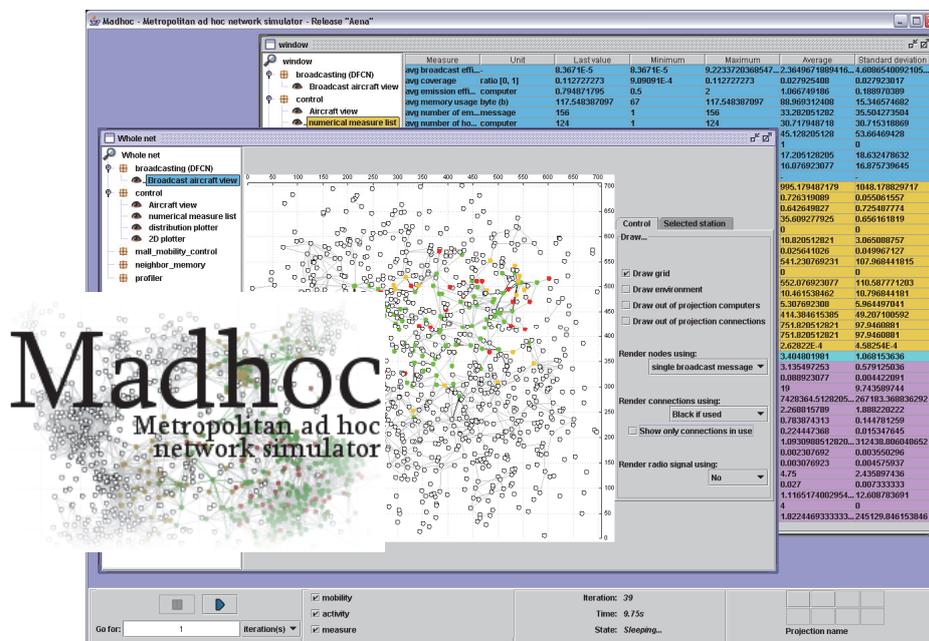


Figura 30. Simulador de redes ad-hoc metropolitanas *Madhoc*.

### 5.3.2 Escenario del problema

El escenario de MANETs metropolitanas elegido para abordar el problema del *broadcasting* se enfoca en centros comerciales, en los cuales las tiendas están situadas normalmente juntas unas a otras en pasillos. La gente va de una tienda a otra a través de estos corredores, parándose de vez en cuando para mirar algunos escaparates. Estos centros comerciales están generalmente muy concurridos (alta densidad de dispositivos), y la gente se comporta de forma diferente (en términos de movilidad) cuando se encuentren dentro o fuera de las tiendas. Generalmente, en este tipo de escenarios existe una alta densidad de tiendas. Por otro lado, en este entorno comercial, tanto la movilidad de los dispositivos como la propagación de su señal están restringidas por las paredes del edificio.

En este contexto, se puede entender que una cadena de centros comerciales desee desplegar un servicio publicitario de comunicación inalámbrica en sus edificios, mediante el cual las personas que lleven sus teléfonos móviles, PDAs, etc., recibirían en sus dispositivos breves anuncios publicitarios, tales como: ofertas especiales, promociones, descuentos, nuevos productos, etc.

### 5.3.3 Protocolo de *broadcasting* "Inundación Retardada con Vecindad Acumulativa"

La estrategia de difusión que se optimiza en esta red es la ya mencionada "Inundación retardada con Vecindad Acumulativa", o *Delayed Flooding with Cumulative Neighborhood (DFCN)* [Hogie 2007]. El motivo de elegir este protocolo de *broadcasting* se debe a que ha sido diseñado específicamente para tratar con las peculiaridades de las MANET metropolitanas.

DFCN requiere el uso de un retraso aleatorio antes de volver a emitir un mensaje  $m$ . Este retraso, llamado *Random Assessment Delay (RAD)* o retraso de valoración aleatorio, se realiza para prevenir las colisiones. Para ser más preciso, cuando un dispositivo  $s$  emite un mensaje  $m$ , todos los dispositivos en  $N(s)$  lo reciben al mismo tiempo. De esta forma, lo más probable es que todos ellos reenvíen el mensaje  $m$  simultáneamente, lo cual puede implicar colisiones de red.

El RAD tiene por objetivo retrasar de forma aleatoria la retransmisión del mensaje  $m$ . Como cada dispositivo en  $N(s)$  espera a la expiración de un RAD diferente antes de reenviar el mensaje  $m$ , el riesgo de colisiones es muy reducido. Aquí,  $N(s)$  representa el conjunto de vecinos de un dispositivo  $s$ .

Se han identificado cinco parámetros que determinan el comportamiento de esta estrategia de difusión. Estos cinco valores (una configuración de DFCN) definen las variables de decisión del problema de optimización y caracterizan el espacio de búsqueda del problema de optimización multi-objetivo. A continuación se describen estos cinco parámetros de DFCN:

- **minGain**: es el mínimo valor para retransmisión (*rebroadcasting*). Este es el parámetro más importante para ajustar DFCN. Su rango va desde 0,0 a 1,0. El parámetro *minGain* es utilizado para calcular el umbral mínimo de la transmisión de un mensaje, es decir, la relación entre el número de

dispositivos vecinos que no han recibido el mensaje y el número total de dispositivos vecinos.

- [**lowerBoundRAD**, **upperBoundRAD**]: define el valor RAD (*Random Assessment Delay* - retraso de valoración aleatorio) para la retransmisión en milisegundos. Ambos parámetros toman valores en el intervalo  $[0,0 - 10,0]$ , expresado en milisegundos. En estos dos parámetros, *lowerBoundRAD* es el límite inferior de RAD y *upperBoundRAD* el límite superior.
- **proD**: representa la densidad de la red y su valor está restringido en el intervalo  $[0, 100]$ .
- **safeDensity**: define una densidad segura del umbral cuyo rango comprende desde 0 a 100 dispositivos. Este umbral se refiere al número de dispositivos en un área (vecindad formada por dispositivos conectados entre ellos). El parámetro *safeDensity* representa la densidad por debajo de la cual el protocolo DFCN siempre retransmite.

### 5.3.4 Algoritmos evolutivos multi-objetivo

En este apartado se explican los algoritmos que se han utilizado en esta tesis para resolver este problema. Como se trata de un problema real de optimización complejo, con un número muy elevado de posibles soluciones, se hace muy difícil para resolverlo de forma exacta dentro de un intervalo de tiempo razonable. Por esta razón se ha optado por utilizar dos heurísticas para resolver el problema: NSGA-II y Evolución Diferencial.

Muchos algoritmos populares para resolver problemas de optimización multi-objetivo son algoritmos evolutivos, principalmente debido a su habilidad de encontrar múltiples soluciones que se compensen en una sola ejecución. Los algoritmos evolutivos trabajan sobre una población de soluciones que son manipuladas mediante la aplicación de varios operadores estocásticos (recombinación y mutación) para buscar una solución óptima satisfactoria.

La heurística Evolución Diferencial (**DE**) ha sido elegida en esta tesis porque ha resuelto exitosamente una amplia variedad de problemas de optimización. En este caso concreto, para abordar el problema se ha implementado un algoritmo Evolución Diferencial Multi-objetivo. Por otra parte, el motivo de utilizar también NSGA-II es porque se trata de uno de los más conocidos y exitosos algoritmos utilizados en optimización multi-objetivo.

#### 5.3.4.1 Codificación

Un individuo de la población se define mediante codificación real para representar a los parámetros del problema (*minGain*, *lowerBoundRAD*, *upperBoundRAD*, *proD*, y *safeDensity*), los cuales ya se han descrito anteriormente. Estos cinco parámetros (que representan una configuración de DFCN) definen las variables de decisión del problema de optimización y caracterizan el espacio de búsqueda de este problema de optimización multi-objetivo.

Por tanto, los dos algoritmos evolutivos (NSGA y Evolución Diferencial Multi-objetivo), que se han utilizado en esta tesis para la resolución del problema, utilizan esta codificación real para representar los parámetros.

#### 5.3.4.2 Función objetivo

Los tres objetivos a optimizar son:

- Minimizar el número de transmisiones.
- Maximizar la cobertura de la red.
- Minimizar la duración del proceso de difusión.

La función de *fitness* es una llamada al programa *ExecSimulator* para ejecutar *Madhoc*. Como se ha comentado anteriormente, *Madhoc* es un simulador de redes metropolitanas ad hoc escrito en Java. Tras ejecutar *ExecSimulator* se obtienen tres valores numéricos que se corresponde con la cobertura, el ancho de banda (número de emisiones de paquetes) y el tiempo de *broadcasting* respectivamente.

#### 5.3.4.3 Algoritmo NSGA-II

Como ya se explicó anteriormente, el algoritmo NSGA-II fue propuesto por Deb. Se trata de un algoritmo genético generacional que basa su funcionamiento en generar una población adicional a partir de una población original mediante el uso de los operadores genéticos típicos de selección, cruce y mutación. Entonces, los individuos más prometedores de ambas poblaciones son seleccionados para la siguiente generación de acuerdo con su rango y distancia de *crowding*. Cuando se utiliza para resolver problemas continuos, NSGA-II aplica selección por torneo binario, cruce SBX (*Simulated Binary Crossover*) y mutación real polinomial.

El algoritmo NSGA-II utiliza elitismo y mecanismos para preservar la diversidad. Se puede apreciar su comportamiento en el pseudo-código de la Figura 31.

Supóngase que  $N$  descendientes son creados desde una población padre de tamaño  $N$ , de forma que la población combinada de tamaño  $2N$  formada por la población padre y la población hijo es ordenada en niveles de no dominación. Esto significa que a cada individuo se le asigna un rango que se basa en la definición de dominancia de Pareto. Todos los individuos no dominados tienen rango 1.

Este tipo de métodos permiten al algoritmo converger hacia el frente de Pareto. Además, para los individuos con el mismo rango, se calcula una estimación de densidad denominada **distancia *crowding*** (factor de agrupamiento). Esta estimación está basada en calcular, para cada solución, la distancia media a dos puntos situados a ambos lados del mismo a lo largo de los objetivos. Las soluciones que tengan mayor distancia *crowding* se encuentran en una región menos aglomerada, es decir, dada una solución  $j$ , las soluciones  $j-1$  y  $j+1$  no se encuentran cerca de la solución  $j$  (Figura 32). De esta forma, la distancia *crowding* guía al algoritmo hacia una distribución de soluciones uniforme.

Entre dos soluciones con diferente rango de no dominación, se selecciona la solución con menor rango porque se considera que es mejor. En el caso de que

ambas soluciones pertenezcan al mismo frente de Pareto, se escoge la solución que está localizada en una región con menor aglomeración.

Los individuos de la población combinada de tamaño  $2N$  son seleccionados para ser insertados en una nueva población de tamaño  $N$  siguiendo los criterios que anteriormente se han explicado. Finalmente, el algoritmo itera continuamente hasta encontrar la condición de terminación.

El algoritmo NSGA-II utiliza un operador de cruce con codificación real (*real-coded crossover*) y un operador de mutación.

```
1: Inicializar una población de tamaño N
2:   Generar aleatoriamente una población de tamaño N
3:   Mantener las soluciones padre dentro del espacio de búsqueda
4:   Evaluar los valores objetivos (fitness) de cada individuo
5:   Asignar rango y distancia crowding a cada individuo
6:   FOR i = 1 hasta un número de generaciones
7:     Generar la población siguiente de tamaño N:
8:     DO
9:       Selección mediante un torneo binario
10:      Recombinación
11:      Mutación
12:      Mantener las soluciones hijo en el espacio de búsqueda
13:     WHILE completar población
14:     Para la población padre e hijo de tamaño 2N:
15:       Asignar rango y distancia crowding a cada individuo
16:       Generar el conjunto de frentes no dominados
17:       Añadir soluciones a la siguiente generación,
18:         empezando por el primer nivel
19:   END FOR
```

Figura 31. Pseudo-Código de NSGA-II.

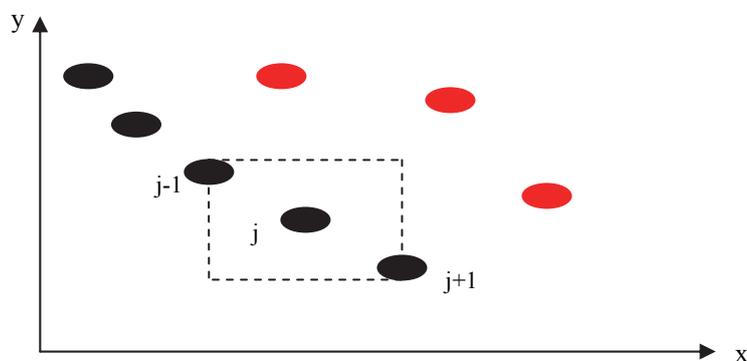


Figura 32. Cálculo de la distancia crowding.

Mantener las soluciones dentro del espacio de búsqueda significa que, si una variable de decisión (parámetro) está fuera de su límite, hay que poner dicha variable igual a su límite correspondiente.

De esta forma, si la variable es menor que su límite inferior, tomará el valor de dicho límite; por el contrario, si una variable es mayor que su límite superior, se le asignará el valor del límite superior.

#### **5.3.4.4 Algoritmo evolución diferencial multi-objetivo**

La Evolución Diferencial (*Differential Evolution*, **DE**) es una heurística que ha demostrado ser exitosa a la hora de abordar problemas de optimización mono-objetivo. Esta tesis extiende la heurística DE para resolver problemas de optimización con más de un objetivo, es decir, una versión multi-objetivo del algoritmo, que llamaremos **Evolución Diferencial Multi-objetivo (EDM)**.

La propuesta EDM modifica NSGA-II de forma que se sustituye el cruce de codificación real y el operador mutación de NSGA-II por el esquema de DE explicado en el apartado 2.6.

El algoritmo DE aplica mecanismos de clasificación de *ranking* basado en Pareto y distancia *crowding* al igual que NSGA-II. Por tanto, esta propuesta utiliza la reproducción del esquema de DE dentro del bucle básico de NSGA-II. Esto es, el algoritmo trabaja con una población de N vectores (array de valores reales) y luego, en cada iteración, se genera una nueva población descendiente de tamaño N mediante el uso del esquema de DE.

Por último, los mecanismos de *ranking* y *crowding* de NSGA-II son utilizados para elegir los N mejores vectores para la próxima iteración.

Del conjunto de estrategias predefinidas, se ha seleccionado la denominada *DE/rand/1/bin*, la cual hace que se mute un vector aleatorio. Por otro lado, se considera un solo vector para la perturbación (del vector aleatorio), y para la recombinación se ha considerado el cruce binomial.

El pseudo-código de EDM basado en NSGA-II se puede ver en la Figura 33.

## 5.4 Resultados experimentales

En este apartado se muestra la parametrización utilizada por cada algoritmo, se describen las métricas usadas para medir la calidad de los frentes de Pareto obtenidos y, finalmente, se analizan los resultados obtenidos en los distintos experimentos que se han realizado.

### 5.4.1 Parametrización

La parametrización del problema es la siguiente:

- La población inicial está formada por 100 individuos.
- La condición de parada de todos los algoritmos consiste en computar 25.000 evaluaciones de DFCN.
- Debido a la naturaleza estocástica del simulador *Madhoc*, se realizan cinco simulaciones por cada función de evaluación y se calcula el valor de *fitness* como la media de los valores obtenidos en cada una de estas simulaciones.
- El tamaño máximo que pueden alcanzar los frentes de Pareto es de 100.

```
1. Inicializar vectores de la población de tamaño N
2. Mantener las soluciones padre dentro del espacio de búsqueda
3. Evaluar el coste de cada vector
4. Asignar rango y distancia crowding a cada vector clasificado
   en un nivel basado en la dominancia de Pareto.
5. FOR i = 1 hasta un número de generaciones
6.   Generar la población siguiente de tamaño N:
7.   DO
8.     Seleccionar (aleatoriamente) 3 vectores diferentes
9.     Realizar mutación usando el esquema Evolución Diferencial
10.    Realizar cruce usando el esquema Evolución Diferencial
11.    Mantener las soluciones hijo dentro del espacio de
   búsqueda
12.    Evaluar los valores objetivos
13.    WHILE la población es completada
14.    Para la población padre e hijo de tamaño 2N:
15.    Asignar rango y distancia crowding a cada vector clasificado
   en un nivel basado en la dominancia de Pareto
16.    Generar el conjunto de frentes no dominados
17.    Añadir soluciones a la siguiente generación, empezando por
   la
   primera jerarquía y usar la distancia crowding en cada
   frente
18.    Seleccionar las soluciones en el frente más bajo (soluciones
   con menor rango) y que estén fuera de la distancia crowding
19. END FOR
```

Figura 33. Evolución Diferencial Multi-objetivo basado en el algoritmo NSGA-II.

La configuración para cada algoritmo es la siguiente:

#### 5.4.1.1 NSGA-II

- A los índices de distribución para el cruce SBX y la mutación polinomial se les ha establecido un valor de 15.
- La selección de los padres se realiza mediante torneo.
- La recombinación utiliza el operador de cruce SBX con probabilidad 0,9. Este operador simula el cruce en un solo punto (*single-point crossover*).
- Por último, el operador de mutación usado en este caso es la mutación polinomial real (*real polynomial mutation*) con probabilidad 0,2. Dicho operador se caracteriza porque utiliza la distribución de probabilidad polinomial para perturbar a las variables.

#### 5.4.1.2 EDM

- Del conjunto de estrategias predefinidas, se ha seleccionado la denominada *DE/rand/1/bin*. Esta estrategia conlleva la mutación de un vector aleatorio, que un único vector es considerado para la perturbación de este vector escogido aleatoriamente, y que para la recombinación se utiliza el cruce binomial.
- El factor de peso F se ha seleccionado como 0,5.
- La constante de cruce C se ha seleccionado como 1,0.

### 5.4.2 *Indicadores de calidad*

A la hora de medir la calidad de algoritmos multi-objetivo se suelen tener en cuenta dos aspectos:

- Minimizar la distancia del frente de Pareto obtenido por el algoritmo al frente de Pareto exacto del problema.
- Maximizar la extensión de soluciones sobre el frente de forma que la distribución sea lo más uniforme posible.

En esta tesis se ha utilizado la métrica **hipervolumen (HV)** [Zitzler 1999], que está diseñada para medir ambos aspectos (convergencia y diversidad) en un frente dado. Esta métrica calcula el volumen (en el espacio de objetivos) cubierto por miembros de un conjunto dado,  $Q$ , de soluciones no dominadas para problemas donde todos los objetivos han de ser minimizados.

Matemáticamente, para cada  $i \in Q$  se construye un hipercubo  $v_i$  con un punto de referencia  $W$  y la solución  $i$  que definen la diagonal del mismo. El punto  $W$  se puede obtener simplemente con los peores valores de las funciones objetivo. Entonces, la unión de todos los hipercubos es lo que define el hipervolumen, tal como se aprecia en la ecuación (25).

$$(25) \quad HV = \text{volumen} \cup_{i=1}^{|Q|} v_i$$

Los algoritmos que alcanzan mayores valores para HV son mejores.

Es necesario normalizar las soluciones no dominadas puesto que HV depende del escalado de los valores de la función objetivo. La normalización realizada en los frentes está basada en calcular los valores máximos y mínimos que se encuentran en todos los frentes resultantes de todos los algoritmos para, a continuación, hacer corresponder todos los valores al intervalo  $[0,1]$ .

No obstante, en esta tesis también se han incluido, junto con los resultados, el número de soluciones no dominadas que los algoritmos son capaces de encontrar, como medida de la capacidad de éstos para explorar espacios de búsqueda altamente complejos, como ocurre en el caso de DFCN.

### 5.4.3 Experimentos

Al trabajar con algoritmos estocásticos, es necesario aportar una cierta confianza estadística a los resultados. Para ello, se ha realizado un análisis estadístico [Demšar 2006], como se explica a continuación.

En la Figura 34 se observa la estrategia utilizada. En primer lugar, se ha aplicado un test de Kolmogorov-Smirnov para determinar si los valores obtenidos siguen una distribución normal (gaussiana). Si la siguen, se utiliza el test de Levene para comprobar la homogeneidad de las varianzas. Si este test es positivo (las varianzas son iguales), se realiza un test ANOVA; en caso contrario, se aplica el test de Welch. Por el contrario, para distribuciones no gaussianas se emplea el test no paramétrico de Kruskal-Wallis, que compara las medianas de los resultados.

El nivel de confianza en todos los tests es del 95% (es decir, un nivel de significancia del 5% o *p-value* por debajo de 0,05), lo cual significa que hay una probabilidad el 95% de que las diferencias hayan ocurrido por casualidad.

Finalmente, y tras aplicar este test, se ha verificado que existe diferencia estadística entre los valores incluidos en la Tabla 8.

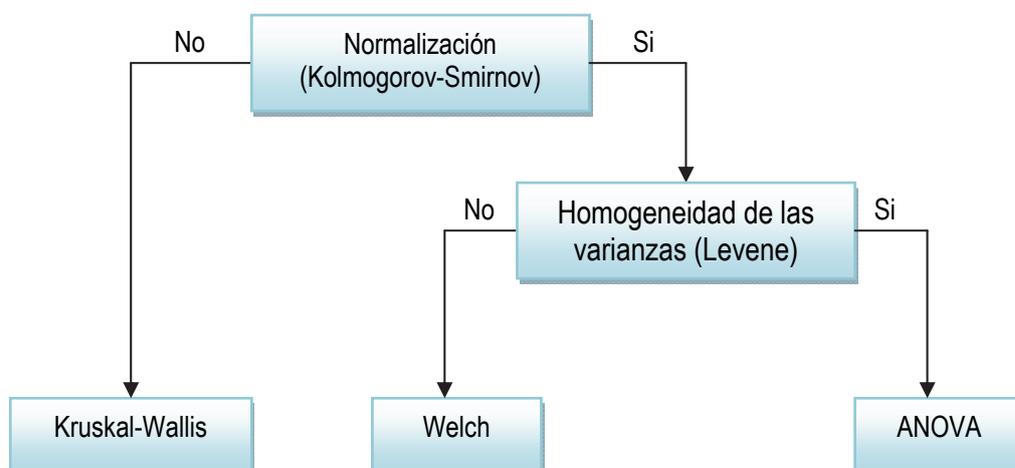


Figura 34. Análisis estadístico de los resultados.

Se han realizado 30 ejecuciones independientes de cada algoritmo. Los resultados se muestran en la Tabla 8, que incluye la mediana  $\tilde{x}$  como medida de localización (o tendencia central).

Algoritmo	$\tilde{x}$
<b>NSGA-II</b>	0,8741
<b>DEM</b>	0,8672

Tabla 8. Valores obtenidos del hipervolumen para los dos algoritmos multi-objetivo.

A partir de los valores de esta tabla se puede observar que el algoritmo NSGA-II obtiene el mejor valor de la métrica, lo cual indica que es la metaheurística más prometedora para resolver el problema de DFCN. De todos modos, las diferencias entre los valores de las métricas no son grandes, lo que indica que el algoritmo Evolución Diferencial Multi-objetivo calcula frentes similares.

El tiempo necesario para ejecutar cada experimento ha sido del orden de dos días y medio. Las ejecuciones se han realizado en un cluster Intel Xeon 3 Ghz con sistema operativo Scientific Linux realease 4.2. Se han realizado 25.000 evaluaciones de la función, que es un valor típico para medir el rendimiento de los algoritmos multi-objetivos que utilizan benchmarks estándares.

No obstante, este número de evaluaciones es probablemente insuficiente para obtener frentes de soluciones precisos en un problema como DFCN. Sin embargo, si se aumenta el número de evaluaciones, el tiempo de ejecución de los algoritmos se incrementará notablemente y resultaría poco práctico, por lo que una posible vía para mejorar los resultados es utilizar técnicas de computación paralela.

#### **5.4.4 Técnicas paralelas aplicadas a algoritmos evolutivos multi-objetivos**

El enfoque paralelo propuesto pretende reducir el gasto de recursos cuando se usan algoritmos evolutivos multi-objetivos. El modelo basado en isla [Veldhuizen 2003] del algoritmo evolutivo paralelo consiste en que la población se divide en un número de sub-poblaciones independientes. Cada sub-población está asociada a una isla. Además, todas las islas están configuradas con idénticos parámetros para que, en cada una de ellas, se ejecute el mismo algoritmo evolutivo.

Cada isla evoluciona de forma aislada al ejecutar el algoritmo para una sub-población dada, pero de vez en cuando se puede permitir que algunas soluciones puedan ser migradas entre islas vecinas, intercambiándose soluciones.

#### 5.4.4.1 Características

Cada procesador disponible constituye una isla. El número de islas y el algoritmo evolutivo multi-objetivo a ejecutar sobre las poblaciones son definidos por el usuario. La población de individuos inicial es generada de forma aleatoria.

Una configuración consiste en el algoritmo evolutivo multi-objetivo junto con el conjunto de parámetros que se define en cada algoritmo, por ejemplo, el índice de la mutación y cruce, el tamaño de la población, etc.

La solución se obtiene mediante la selección de las soluciones no dominadas de los obtenidos por las islas. La solución global tiene un tamaño limitado. Para conseguir esto, se aplica el operador *crowding* NSGA-II; de esta forma se limita el tamaño del conjunto de la solución global.

En este modelo de isla también se define el criterio de parada para la ejecución de los algoritmos en las islas. De esta forma, cuando se alcanza un criterio de parada, la ejecución de la isla se da por finalizada.

#### 5.4.4.2 Experimentos

Se han utilizado los mismos algoritmos evolutivos explicados en los anteriores apartados. La parametrización para cada uno de ellos ha sido la siguiente:

- Configuración de parámetros para el algoritmo NSGA-II:
  - Índice de distribución para cruce SBX (*Simulated Binary Crossover*) = 20.
  - Índice de mutación polinomial = 20.
  - Probabilidad de cruce SBX = 0,9.
  - Probabilidad de mutación = 0,2.
- Configuración de parámetros para el algoritmo Evolución Diferencial Multi-objetivo:
  - Estrategia: ED/aleatorio/1/bin.
  - Factor de peso = 0,5.
  - Constante de cruce = 1,0.

La probabilidad de migración es de 0,05 y el número de individuos a migrar está limitado a 4 cada vez. Por otro lado, el criterio de parada para cada ejecución fue fijado en 25.000 evaluaciones. La solución final estaba limitada a 100 elementos.

Para realizar las pruebas basadas en la propuesta paralela del modelo de islas, la población se forma a partir de 8 sub-poblaciones (islas), y cada una de ellas está formada a su vez por 15 individuos.

Para dar validez estadística a los resultados, cada experimento se ha ejecutado 30 veces para obtener valores medios.

Finalmente, y con el propósito de aportar resultados con confianza, se han realizado los mismos análisis estadísticos comentados en el apartado anterior. De esta forma, en primer lugar se hace una prueba de Kolmogoriv-Smirnov para

comprobar si el resultado sigue una distribución normal gaussiana. Después, cada prueba pasa el test de normalidad. A continuación, la homogeneidad de las varianzas para cada par de pruebas se asegura a través del test Levene. Por último, el test ANOVA se ejecuta para comprobar los niveles de confianza.

En la Tabla 9 se muestra el hipervolumen medio logrado por cada modelo paralelo en los límites dados. El indicador de hipervolumen hace posible combinar la información de calidad de convergencia y la diversidad en un único valor.

Algoritmo	$\tilde{x}$
<b>NSGA-II paralelo</b>	0,739
<b>DEM paralelo</b>	0,727

Tabla 9. Valores obtenidos del hipervolumen para los algoritmos paralelos.

Las ejecuciones paralelas se desarrollaron en un entorno hardware / software de las siguientes características:

- Hardware: Cluster Intel Xeon 3 GHz.
- Sistema operativo: Scientific Linux release 4.2.
- MPI: MPICH2 version 1.0.4.
- Compilador C: gcc 3.3.

#### 5.4.5 Comparativa con heurísticas de otros autores

Para conocer el grado de efectividad de las soluciones halladas utilizando NSGA-II y DEM, tanto en sus implementaciones secuencial y paralela, es necesario comparar los resultados con los obtenidos por otros autores. En este caso, la comparación ha sido fácil, ya que este trabajo se enmarca en un proyecto de investigación colaborativo, OPLINK, donde investigadores de distintas universidades abordaron el mismo problema utilizando distintas heurísticas.

##### 5.4.5.1 Proyecto OPLINK

El proyecto OPLINK<sup>3</sup> es un proyecto de investigación financiado por el Ministerio de Educación y Ciencia para los años 2005 al 2007 dentro del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica.

El proyecto OPLINK, que lleva por título "*Net Centric Optimization / Optimización y Ambientes de Red*", es un proyecto coordinado, formado por subproyectos liderados por los grupos que se enumeran a continuación:

- OPLINK::UMA. Investigadores del Grupo GIISUM, de la Universidad de Málaga (Coordinador).

<sup>3</sup> <http://oplink.unex.es>

- OPLINK::UEX. Investigadores del Grupo de Arquitectura de Computadores y Diseño Lógico y del Grupo de Evolución Artificial, de la Universidad de Extremadura.
- OPLINK::UC3M. Investigadores del Grupo ScaLab de la Universidad Carlos III de Madrid.
- OPLINK::ULL. Investigadores del Grupo de Computación Paralela de la Universidad de La Laguna.

Así, el trabajo sobre MANET desarrollado en esta tesis fue sufragado por el subproyecto OPLINK::UEX, que tiene como identificador TIN2005-08818-C04-03 (en la Figura 35 puede verse una imagen del portal de Internet de este subproyecto).

La propuesta de OPLINK estaba orientada a resolver problemas de optimización en el campo de las redes y comunicaciones (ambientes de red). Para ello, se propuso utilizar métodos exactos, heurísticos y, en general, cualquier técnica novedosa que permita su resolución de forma eficiente y precisa. Dentro de estas aplicaciones orientadas a red se incluyó el diseño de redes ad-hoc, así como la utilización de la computación paralela.

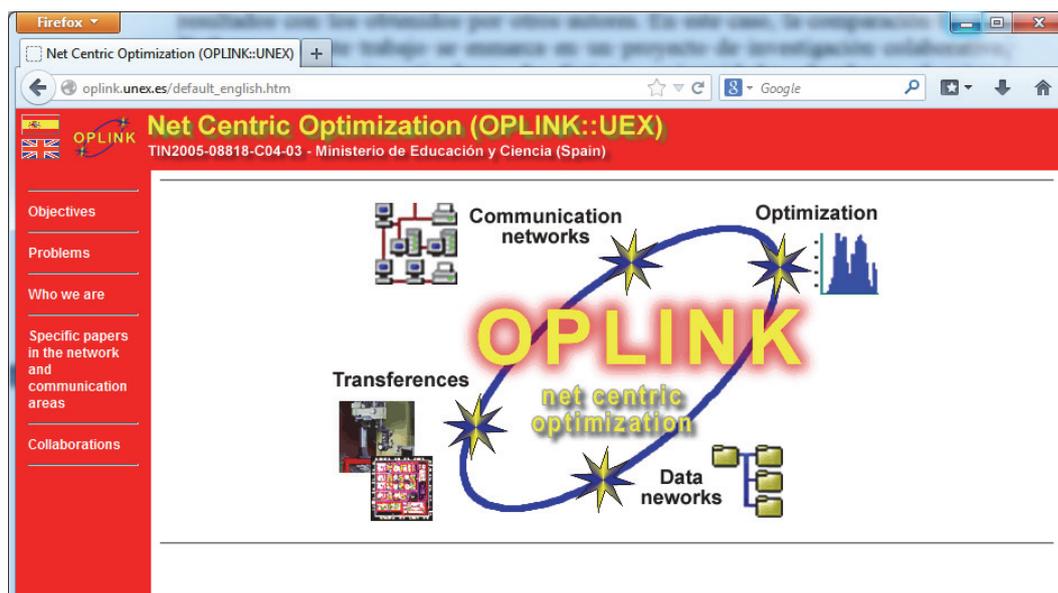


Figura 35. Portal en Internet del proyecto OPLINK: <http://oplink.unex.es>.

#### 5.4.5.2 Análisis comparativo

Los diversos grupos de investigación que abordaron este problema en el marco del proyecto OPLINK utilizaron otras heurísticas de optimización multi-objetivo, cuyos resultados se pueden comparar con los aportados en este trabajo.

Básicamente, la comparación que interesa es respecto a DEM, que es el propuesto en este trabajo, ya que el algoritmo NSGA-II fue implementado por todos los grupos de forma común (los resultados de NSGA-II que se exponen aquí son los obtenidos en este trabajo).

En la Tabla 10 puede observarse esta comparativa. Se deduce que el algoritmo DEM tiene un comportamiento razonablemente bueno, comportándose mejor que MOPSO y CMOGA y prácticamente igualando a AbYSS.

Algoritmo	$\tilde{x}$
<b>SPEA2</b>	0,8807
<b>ES</b>	0,8755
<b>NSGA-II</b>	0,8741
<b>AbYSS</b>	0,8687
<b>DEM</b>	0,8672
<b>MOPSO</b>	0,8644
<b>CMOGA</b>	0,8480

Tabla 10. Valores obtenidos del hipervolumen para los algoritmos multi-objetivo aplicados en el proyecto OPLINK, ordenados de mejor a peor resultado .

En la Tabla 11 se aprecia otra comparativa, esta vez para las implementaciones paralelas de los algoritmos de optimización. Sigue la tónica de un rendimiento razonable de DEM, mejorando ahora su posición en el ranking de algoritmos.

Algoritmo	$\tilde{x}$
<b>NSGA-II paralelo</b>	0,739
<b>SPEA2</b>	0,738
<b>ADAP-IBEA</b>	0,733
<b>IBEA</b>	0,733
<b>DEM paralelo</b>	0,727
<b>ESN</b>	0,726
<b>MOCcell</b>	0,712
<b>MOPSO</b>	0,685

Tabla 11. Valores obtenidos del hipervolumen para los algoritmos multi-objetivo paralelos aplicados en el proyecto OPLINK, ordenados de mejor a peor.

## 5.5 Conclusiones

En este trabajo de tesis se han analizado dos metaheurísticas multi-objetivo para resolver el problema del ajuste del protocolo de difusión en redes MANETs metropolitanas. Se trata de un problema complejo del mundo real, que utiliza *Madhoc*, un simulador de red MANET metropolitana, para modelar un escenario realista que representa un centro comercial. Este escenario conduce a un problema de optimización multi-objetivo con cinco variables de decisión y tres objetivos.

Una de las metaheurísticas elegidas es representativa en el campo de la optimización multi-objetivo, como es el algoritmo NSGA-II. Por otro lado, se ha considerado también otra metaheurística multi-objetivo adaptada del algoritmo de Evolución Diferencial. Los experimentos realizados revelan que NSGA-II es el algoritmo que produce los mejores resultados, según la métrica hipervolumen. Sin embargo, los valores de las métricas parecen indicar que ambos algoritmos producen unos frentes que están muy próximos.

Para tratar de mejorar los frentes de Pareto del problema y optimizar el rendimiento de los algoritmos, se ha planteado en esta tesis la utilización de técnicas de computación paralela. De esta forma, se pueden calcular más evaluaciones de la función en un plazo razonable de tiempo.

Siguiendo esta línea, esta tesis ha definido una propuesta de optimización paralela multi-objetivo para el mismo problema de optimización, basado en un enfoque paralelo según el modelo de islas. Este enfoque paralelo ha permitido conseguir resultados óptimos minimizando el tiempo y consumo computacional. Por tanto, los resultados logrados para este escenario demuestran que esta técnica puede ser aplicada a otros escenarios MANET.

Los resultados del hipervolumen obtenidos por DE han sido razonablemente buenos cuando se comparan con los aportados por otras heurísticas de optimización multi-objetivo: si bien no es la mejor heurística, está en la línea de buenos resultados y mejora los de otras heurísticas, especialmente en la implementación paralela.

Como posibles líneas de trabajo futuro se pueden plantear:

- Abordar otros protocolos de *broadcasting* para resolver las mismas instancias del problema de optimización.
- Definir otros escenarios MANETs.
- Aplicar nuevos algoritmos evolutivos de optimización multi-objetivo.
- Utilizar otras métricas para medir la calidad de los algoritmos ejecutados.

## 5.6 Aportaciones

Como aportaciones personales realizadas por la doctoranda en este problema de optimización del ajuste del protocolo de difusión en redes MANETs, cabe citar:

- Modelado del problema para su ejecución por los algoritmos NSGA-II y EDM.
- Ejecución de experimentos, obtención de resultados a partir de la métrica hipervolumen y análisis de datos mediante tests estadísticos.
- Modelado y programación paralela de los algoritmos de optimización.

Por otro lado, cabe destacar que la experiencia obtenida en el algoritmo de Evolución Diferencial ha posibilitado la colaboración de la doctoranda en algunos aspectos de otra tesis doctoral, similar en cuanto a ámbito (redes de comunicaciones) y algoritmos (Evolución Diferencial), aportando asesoramiento en las características funcionales del algoritmo, especialmente en sus estrategias.

En cuanto a la productividad científica, las anteriores aportaciones aparecen en las siguientes publicaciones:

- **Capítulos de libro.**
  - "Optimizing the DFCN Broadcast Protocol with a Parallel Cooperative Strategy of Multi-Objective Evolutionary Algorithms", in: "Evolutionary Multi-Criterion Optimization". Carlos Segura, Alejandro Cervantes, Antonio J. Nebro, María D. Jaraíz, Eduardo Segredo, Sandra García, Francisco Luna, Juan A. Gómez, Gara Miranda, Cristóbal Luque, Enrique Alba, Miguel A. Vega, Coromoto León, Inés M. Galván. Springer-Verlag, Berlin, Germany, 2009, pp.:305-319. ISBN:978-3-642-01019-4.
  - "Metaheuristic Approaches for Optimal Broadcasting Design in Metropolitan MANETs", in: "Computer Systems Aided Theory". E. Alba, A. Cervantes, J.A. Gómez, P. Isasi, M.D. Jaraíz, C. León, C. Luque, F. Luna, G. Miranda, A.J. Nebro, R. Pérez, and C. Segura. R. Moreno-Díaz et al. (Eds.), Springer-Verlag, Berlin Heidelberg, Germany, 2007, pp.:755-763. ISBN: 978-3-540-75866-2.
- **Conferencias.**
  - "Parallel Differential Evolution for Multi-objective Optimization applied to the DFCN Broadcasting Strategy in MANETs". María D. Jaraíz-Simón, Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez, Juan M. Sánchez-Pérez. Computer Aided Systems Theory - Extended Abstracts, A. Quesada, J. Rodriguez, R. Moreno. IUCTC. Universidad de Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain, 2009, pp.:238-239. ISBN: 978-84-691-8502-5.
  - "Algoritmo basado en Evolución Diferencial para el Problema de Redes de Radiocomunicaciones". J. A. Gómez Pulido, S. Mendes Priem, M. D. Jaraíz Simón, Miguel A. Vega-Rodríguez, J. M. Sánchez Pérez. Actas del V Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y

Bioinspirados, F. Almeida, B. Melián, J. Moreno, M. Moreno. Universidad de La Laguna, España, 2007, pp.:309-314. ISBN: 978-84-690-3470-5.

- "Metaheuristics approaches for optimal broadcasting design in metropolitan MANETs". E. Alba, A. Cervantes, J.A. Gomez, P. Isasi, M.D. Jaraíz, C. Leon, C. Luque, F. Luna, G. Miranda, A. J. Nebro, R. Perez and C. Segura. Computer Aided Systems Theory - Extended Abstracts, A. Quesada, J. Rodriguez, R. Moreno. IUCTC. Universidad de Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain, 2007, pp.:262-263. ISBN: 978-84-690-3603-7.
- "A Differential Evolution Based Algorithm to Optimize the Radio Network Design Problem". Silvio Priem Mendes, Juan A. Gómez Pulido, Miguel A. Vega-Rodríguez, María D. Jaraíz Simón, Juan M. Sánchez Pérez. Proceedings of the Second IEEE International Conference on e-Science and Grid Computing, IEEE, Amsterdam, The Netherlands, 2006, pp.:119-125. ISBN: 0-7695-2734-5.

# ***6 Optimización de la fase de decisión en el proceso de Vertical Handoff para terminales móviles en redes inalámbricas heterogéneas***

*“A veces sentimos que lo que hacemos es tan solo una gota en el mar, pero el mar sería menos si le faltara una gota”.*

**Madre Teresa de Calcuta.**  
Religiosa y misionera albanesa,  
nacionalizada india.  
Beatificada en 2003 por  
Juan Pablo II (1910-1997)

## 6.1 Resumen

En este capítulo se aborda un **problema de optimización en el ámbito de las comunicaciones móviles sobre redes inalámbricas heterogéneas**, donde se considera la **calidad del servicio** como la principal característica a considerar al establecer una conexión. Este problema se enmarca en una de las cinco líneas de investigación actuales más importantes en la llamada “Internet 3.0” [Pan 2011].

Resolver este problema de forma **rápida** y con un **bajo coste computacional** es indispensable debido a la restricción impuesta por los microprocesadores de bajas prestaciones que mayoritariamente albergan muchos de los dispositivos móviles que establecen las conexiones (tablets, smartphones, etc.). Por tanto, las heurísticas que aporten una buena solución al problema tienen que ajustarse obligatoriamente a estas dos restricciones: rapidez en suministrar la solución (cercanía al tiempo real) y ligereza del algoritmo (que pueda ser ejecutado en microprocesadores empotrados con escasa capacidad de memoria de programa).

El problema abordado en esta tesis tratará con redes inalámbricas de **tecnología heterogénea** (UMTS, WiMax, WLAN, etc.), capaces de proporcionar diversos servicios (voz, datos), y que se considerarán como puntos de acceso a la red para terminales que necesitan mantener la conectividad en todo momento.

Cada red se caracteriza por una serie de parámetros que proporcionan la llamada **calidad del servicio (QoS)**, como ancho de banda, tiempo de respuesta, coste económico, nivel de seguridad, etc. Sus valores indican la calidad de una red para ofrecer al usuario un servicio determinado (por ejemplo, para transmisión de datos es más importante una baja tasa de errores de transmisión de bits que el tiempo de respuesta, el cual en cambio es más importante para otros servicios, como las conversaciones de voz).

Las redes inalámbricas en este problema son usadas como puntos de acceso a la red para un **terminal móvil**. Así, se habla de **entornos dinámicos** donde hay que tener en cuenta la movilidad del terminal a la hora de desarrollar algoritmos que solucionen el problema, pues la obtención de soluciones óptimas ha de ser rápida. Esta es una característica distinta a otros problemas de optimización en el campo de las comunicaciones inalámbricas, como pueden ser el posicionamiento óptimo de nodos, la arquitectura de redes, la asignación de bandas de frecuencia, problemas que permiten la ejecución de algoritmos que obtienen soluciones óptimas al cabo de muchas horas.

Este problema de optimización surge a partir de la necesidad de ofrecer **conexión ininterrumpida de máxima calidad** a terminales móviles que pueden desplazarse a gran velocidad (dispositivos a bordo de vehículos o aeronaves) en escenarios donde hay presentes varias redes de distinta tecnología. Cuando el terminal descubre nuevas redes en su desplazamiento, ha de evaluar la calidad de servicio que proporcionan con el fin de tomar la decisión de si abandona la actual conexión o no, y en caso afirmativo decidir a qué nueva red conectarse: a este proceso se le denomina **Vertical Handoff (VH)** [Kassar 2008].

La fase de decisión VH ha de ejecutarse nuevamente al descubrirse una nueva red (o si varían los parámetros de la actual), cosa frecuente en entornos dinámicos, de ahí la necesidad de encontrar algoritmos de decisión rápidos.

Para una red dada, su conjunto de parámetros QoS posee ciertos valores; el peso normalizado que se le asigne a cada uno de ellos permite obtener una medida global de calidad de la red, constituyendo una posible solución. De entre todas las posibles soluciones (combinaciones de pesos), la óptima será aquella combinación que ofrezca un mejor valor de la medida de calidad de esa red.

Si se evalúa este proceso de optimización para todas las redes que el terminal puede ver, se podrá determinar cuál es la solución óptima global (la mejor solución de entre las mejores soluciones de todas las redes) y, por tanto, cuál es la red a la que el terminal va a conectarse.

Por tanto, el problema de optimización consistirá en, para cada red, buscar la solución óptima de entre todas las existentes en el espacio de soluciones y, una vez hecho, determinar la mejor de las soluciones óptimas para escoger la red a la que conectarse. Ante posibles espacios de soluciones muy grandes, que aumentan al contemplar un mayor número de parámetros QoS, y tener que obtenerse soluciones óptimas en tiempo real, las búsquedas directas estarán claramente descartadas, debiéndose recurrir a heurísticas rápidas de búsqueda no exhaustiva.

Es importante tener en cuenta que los terminales móviles ejecutan los algoritmos de VH con microprocesadores de bajas prestaciones, generalmente de un solo núcleo y pocos centenares de MHz, dadas las necesidades de bajo consumo y alta integración que han de tener los microprocesadores empotrados en dispositivos móviles (tabletas, teléfonos, etc.), por lo que el desafío de abordar este problema de optimización aumenta al ser necesario implementar heurísticas ligeras (código de poco tamaño que ocupe poca memoria) y rápidas (que puedan proporcionar soluciones en tiempos pequeños, incluso con instancias del problema que impliquen un elevado coste computacional).

Los trabajos de optimización de esta tesis parten del desarrollo de una propuesta novedosa de algoritmo de búsqueda directa (que evalúa todas las posibilidades) no exhaustiva (para sub-espacios de los espacios posibles de soluciones) que tiene un coste computacional proporcional a la precisión dada para la búsqueda. Este algoritmo ha sido utilizado para determinar el coste computacional y complejidad del problema de optimización y para integrarse en dos propuestas de heurísticas evolutivas, necesarias para permitir la obtención de buenas soluciones sin elevar el tiempo de computación, cosa que no puede evitar dicho algoritmo.

La resolución eficiente de este problema es muy relevante en el ámbito de las redes inalámbricas para muchas aplicaciones avanzadas que necesitan ofrecer al usuario una alta calidad del servicio. De esta forma, los resultados obtenidos en esta tesis permiten proponer las heurísticas desarrolladas como algoritmos útiles para ejecutar la fase de decisión del *Vertical Handoff* en dispositivos móviles bajo redes inalámbricas heterogéneas.

## 6.2 Introducción

### 6.2.1 ¿Qué es Vertical Handoff?

Cuando un terminal móvil cambia de conexión de una estación base (BS) a otra (actuando ambas como puntos de acceso a la red, tal como se puede ver en la Figura 36), puede producirse una de estas dos situaciones [Kassar 2008]:

- *Vertical Handoff* (VH) ó *intersystem handoff*: Cuando la conexión del terminal móvil salta de una red a otra.
- *Horizontal Handoff* (HH) ó *intrasystem handoff*: Cuando la conexión del terminal móvil salta de una BS a otra dentro de una misma red.

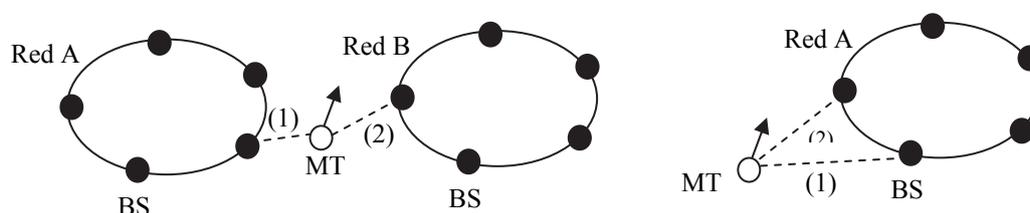


Figura 36. Vertical Handoff (izquierda) y Horizontal Handoff (derecha).

Los terminales móviles tienen interfaces hardware múltiples para establecer conexiones con distintos tipos de redes, dada la naturaleza heterogénea de muchos escenarios de conectividad inalámbrica.

El estudio del presente trabajo de optimización se centra en el proceso VH, el cual consta de las siguientes tres fases:

- **Descubrimiento.** Dentro del área de cobertura del terminal móvil hay varias redes. El terminal determina qué redes pueden usarse y qué servicios dispone cada una. Esta fase es periódicamente invocada, ya que en su trayectoria el terminal puede perder visibilidad de redes actuales y descubrir otras nuevas.
- **Decisión.** El terminal móvil determina tanto si la conexión debe seguir en la misma red o si salta a otra. La decisión dependerá de varios parámetros relacionados con la calidad del servicio (tipo de aplicación, ancho de banda mínimo, coste de acceso, potencia transmitida, tiempo de batería, etc.).
- **Ejecución.** En esta fase las conexiones son encaminadas nuevamente de la actual red a otra, para lo cual se requiere autenticación, autorización, transferencia de información, etc.

Cada una de estas fases tiene sus propios algoritmos y procedimientos. La primera y tercera necesitan interactuar con los dispositivos de entrada y salida del terminal (antenas, conversores A/D y D/A,) y con los distintos circuitos de comunicaciones, mientras que la segunda fase es un proceso eminentemente algorítmico, que solo necesita de la computación de código.

Es en esta **fase de decisión** en la que se ha centrado el problema de optimización, dado que requiere únicamente de procesamiento algorítmico.

### 6.2.2 Trabajos relacionados

Existen muchas propuestas algorítmicas para la fase de decisión VH en redes inalámbricas heterogéneas, que cubren diferentes aspectos de las comunicaciones. El aspecto negativo de la existencia de tantas propuestas distintas es que se dificulta la posibilidad de hacer un estudio homogéneo comparativo de rendimientos y eficacias en la selección de la mejor red al aplicar distintos algoritmos. Por ejemplo, [Lee 2009] propone un algoritmo de decisión VH que busca un balance eficiente de la carga global de comunicación entre las estaciones base y los puntos de acceso, maximizando el tiempo de vida de las baterías de los terminales móviles, mientras que [He 2011] propone un algoritmo sencillo de dos fases que se centra en obtener una medida de la robustez requerida por los terminales móviles que usan baterías, teniendo en cuenta la cantidad de recursos proporcionados por las redes.

Estas y otras propuestas algorítmicas ofrecen buenos comportamientos pero bajo ciertas condiciones y teniendo en cuenta aspectos a veces muy distintos de las comunicaciones inalámbricas. En otras palabras, cada algoritmo centra su enfoque en aspectos distintos que no son considerados, o lo hacen parcialmente, por otros. No obstante, el interés investigador de esta tesis se centra específicamente en las características de calidad del servicio de las comunicaciones.

En este sentido, existen muchos algoritmos para la fase de decisión VH que basan sus comportamientos en los parámetros QoS proporcionados por las redes inalámbricas de naturaleza heterogénea. De esta forma, en [Yan 2010] se muestra un amplio estudio de algoritmos de decisión diseñados para proporcionar la calidad de servicio requerida para un amplio rango de aplicaciones. Estos algoritmos son agrupados en cuatro categorías según el criterio de decisión utilizado: parámetro RSS (*received signal strength*) [Zahran 2005] [Mohanty 2006] [Yan 2008], ancho de banda [Lee 2005] [Yang 2007] [Chi 2007], función de coste o fitness [Zhu 2004] [Hasswa 2006] [Tawil 2008], o combinación de parámetros [Nasser 2007] [Pahlavan 2000] [Xia 2007]).

Adicionalmente, en [Bhuvaneswari 2012] se expone una revisión de algoritmos de decisión VH, agrupándolos como: algoritmos tradicionales, basados en funciones, centrados en el usuario, atributos múltiples, lógica borrosa y redes neuronales, y sensibles al contexto.

Otra línea interesante de investigación explota la técnica llamada proceso de decisión de Markov, *Markov Decision Process (MDP)*, para formular los mecanismos de decisión VH. Así, [Stevens 2008] propone un algoritmo MDP con el objetivo de maximizar la llamada recompensa de una conexión teniendo en cuenta la calidad del servicio, y [Sharna 2011] propone dos algoritmos mejorados de tipo MDP (MDP\_SAW y MDP\_TOPSIS) para obtener la mejor red disponible en términos de ancho de banda.

No obstante, estos algoritmos de decisión VH no suelen manejar muchos parámetros QoS; normalmente consideran el parámetro RSS junto con algún otro, o pocos más. Por ejemplo, en [Lee 2006] se propone un algoritmo de decisión basado en la llamada función de utilidad utilizando la capacidad de Shannon, pero considerando únicamente un solo parámetro QoS: la productividad. Más aún, en [Ceken 2010], un algoritmo basado en lógica borrosa que considera perfiles de usuario, requisitos de la aplicación y condiciones de la red, se limita a dos parámetros QoS: tasa de transferencia de datos y RSS; esto lo hace para centrar la atención sobre los condicionamientos de las interferencias. Otro ejemplo puede verse en [Vasu 2012], donde se consideran cuatro parámetros QoS (ancho de banda, retardo final, jitter y tasa de error de bit) que son tratados por un algoritmo de decisión tipo MCDM (que se define en el siguiente párrafo) y que utiliza también lógica borrosa.

El interés de la investigación realizada en esta tesis reside en considerar metodologías capaces de manejar un número alto de parámetros QoS, evitando incrementar demasiado los tiempos de computación. En esta línea, la mayoría de los algoritmos consideran un conjunto de pesos asociados a los parámetros QoS como la base para efectuar la decisión VH. Estos algoritmos buscan una óptima selección de pesos para los parámetros QoS que minimicen una función de fitness capaz de proporcionar el grado de bondad de cada red disponible en un momento determinado. De esta forma, se encuentra un tipo concreto de algoritmos de decisión llamados **Multiple Criteria Decision-Making (MCDM)**.

Entre los algoritmos de tipo MCDM destacan los algoritmos *Analytic Hierarchy Process (AHP)* [Song 2005], *Simple Additive Weighting (SAW)* y *Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)* [Lassoued 2008]. Son algoritmos sencillos, que utilizan reglas de poca complejidad para determinar la combinación de pesos QoS. En esta línea también hay aproximaciones inteligentes como la propuesta en [Nkansah 2007], donde un algoritmo de lógica borrosa de tipo MCDM es capaz de seleccionar hasta 9 parámetros QoS.

Los algoritmos de decisión VH propuestos en esta tesis están capacitados para considerar varios parámetros QoS, e implementan técnicas que manejan de forma eficiente los pesos asociados. Por estas razones, y condicionados por la inexistencia de instancias experimentables comunes y la dificultad de reproducir los experimentos y programación de los algoritmos a partir de la información aportada en los artículos publicados, las comparativas de rendimiento y eficacia realizadas en esta tesis se limitan a los algoritmos de tipo MCDM.

### **6.2.3 Parámetros de calidad de servicio para la fase de decisión**

En los procesos tradicionales de VH solamente se consideraban, como características de calidad de servicio, la disponibilidad del canal y la fuerza de la señal. Sin embargo, para las redes de nueva generación 3G y 4G, presentes en multitud de ámbitos actuales, se contemplan otros importantes parámetros que afectan notablemente a la calidad del servicio, proponiéndose, entre otros, los siguientes [McNair 2004][Nkansah 2007]:

- Tipo de servicio.
- Coste monetario.
- Condiciones de la red:
  - Tráfico.
  - Ancho de banda disponible.
  - Latencia.
  - Congestión (pérdida de paquetes).
  - Fuerza de señal de recepción (RSSI).
  - Cobertura.
- Rendimientos del sistema:
  - Características de propagación del canal.
  - Pérdida de ruta.
  - Tasa señal-ruido (SNR).
  - Tasa de error de bit (BER).
  - Nivel de energía de la batería.
  - Robustez.
  - Seguridad.
- Condiciones del terminal móvil:
  - Velocidad.
  - Patrón de movimiento.
  - Historial de trayectorias.
  - Información de localización.
  - Potencia de la batería.
- Preferencias del usuario.

#### **6.2.4 Métrica de calidad y función de fitness**

En la fase de decisión de VH han de pesar o tenerse en cuenta los distintos factores de calidad de servicio presentes en las redes inalámbricas heterogéneas.

Una determinada red viene caracterizada por su conjunto de parámetros QoS, los cuales tienen unos valores determinados. En las métricas de calidad de servicio de la red, se considera un conjunto o combinación de pesos asignados a cada uno de estos parámetros para obtener una medida de la calidad de esta red. En esta tesis se denominará como posible solución a una combinación cualquiera de pesos que cumpla cierta restricción (que se verá más adelante).

La métrica de calidad que utilizan estas soluciones o combinaciones de pesos viene dada por la llamada **función de fitness**. Para que se pueda tomar una decisión VH a partir de estas soluciones, ha de evaluarse la "bondad" o grado de calidad de las mismas mediante la función de *fitness*. Esta función, que es evaluada en la fase de decisión VH, mide el grado de rendimiento de la red basándose en diversos factores, como las preferencias del usuario [WangH 1999][Song 2008][McNair 2004].

En el caso de las redes multiservicio (que proporcionan varios servicios de comunicaciones simultáneamente), la función de *fitness* considera dos dimensiones: el tipo de servicio solicitado por el usuario y el *fitness* de la red de acuerdo con parámetros específicos (como ancho de banda, consumo de energía y coste económico). Para simplificar la aproximación al problema de optimización, en los estudios realizados para la tesis se consideran redes monoservicio.

La función de fitness debe ser minimizada o maximizada según distintos enfoques adoptados por varios autores, en cuyo caso la formulación de la función cambia en cuanto a la forma de representar los valores de los parámetros QoS. Así, la función de fitness se minimiza en [WangH 1999] y [McNair 2004] (que denominan como "función de coste"), o se maximiza en [Song 2008] y [Chiasserini 2004] (donde es denominada como "función de mérito"), y en [Nkansah 2007] ("función de fitness"). En este sentido, los términos función de fitness, coste, mérito, etc., indican un mismo objetivo: evaluar la calidad de la red para responder a la decisión de efectuar un VH.

En esta tesis se utiliza la formulación de la función de calidad dada en [WangH 1999], para la cual, en adelante, se utilizará el término **función de fitness F a minimizar**, en lugar de función de coste, porque el término "coste" es preferible reservarlo para referirse al coste económico del servicio, que es un importante parámetro de la calidad del servicio.

Por tanto, se utilizará la siguiente función de *fitness*:

$$(26) \quad F^{(n)} = E^{(n)} \sum_s \sum_i w_{s,i} N(p_{s,i}^{(n)})$$

con la importante restricción:

$$(27) \quad \sum_i w_i = 1$$

es decir, que la suma de pesos siempre ha de ser 1, donde:

$n$  número que identifica a la red.

$E^n$  factor de eliminación.

$s$  número que identifica al servicio.

$i$  número que identifica al parámetro de calidad de servicio.

$w_{s,i}$  peso asignado al parámetro  $i$  para ejecutar el servicio  $s$ .

$p_{s,i}^n$  valor del parámetro de calidad  $i$  para el servicio  $s$ , donde  $N$  es una función de normalización.

La restricción indicada en la ecuación (27) es de suma importancia, ya que va a condicionar enormemente el diseño de la metodología de resolución de este problema de optimización.

Considerando redes mono-servicio (por ejemplo, una red que solo proporciona servicio de datos), y sin contemplar de momento el factor de eliminación, se simplifica el problema y por tanto su función de *fitness*:

$$(28) \quad F^{(n)} = \sum_i w_i N(p_i^{(n)})$$

Por un lado, se suele considerar como factor de normalización al logaritmo neperiano. Por otro lado, según se entienda el valor  $p$ , la función de *fitness* se debe minimizar o maximizar para indicar la calidad de la red. Por ejemplo, si se considera que un valor menor de  $F$  indica una red de mayor calidad, los valores  $p$  de algunos parámetros QoS deben invertirse; por ejemplo, si incrementar el ancho de banda aumenta la calidad de la red, el factor  $p$  correspondiente al ancho de banda ha de invertirse para que su peso en  $F$  la reduzca.

Teniendo en cuenta estos dos aspectos (normalización y minimización de  $F$  como medida de calidad la red), la función de *fitness* queda como:

$$(29) \quad F^{(n)} = \sum_i w_i \ln(p_i'^{(n)})$$

donde:

- se aplica (30) si a mayor  $p$ , empeora la calidad de la red (por ejemplo, retardo y coste económico), amentando por tanto el *fitness*.
- se aplica (31) si a mayor  $p$ , mejora la calidad de la red (ancho de banda), disminuyendo por tanto el *fitness*.

$$(30) \quad p_i'^{(n)} = p_i^{(n)}$$

$$(31) \quad p_i'^{(n)} = \frac{1}{p_i^{(n)}}$$

En esta tesis se considera siempre la función de *fitness* dada en la ecuación (29). Esto se puede ilustrar con dos ejemplos:

Ejemplo 1:

- 2 parámetros:
  - Ancho de banda  $b$ :  $fitness = \ln(1/bn)$ .
  - Coste económico  $c$ :  $fitness = \ln(cn)$ .
- 1 servicio: datos.

Nótese que, según se considere cómo mejora o empeora la calidad de la red, la formulación de la función de *fitness* implica minimizarla o maximizarla. Así por ejemplo, en [WangH 1999] y [McNair 2004] se considera:

- $F^{(n)} = w_b \ln(1/b^{(n)}) + w_c \ln(c^{(n)})$ , donde  $w_b + w_c = 1$

mientras que en [Song 2008] se considera:

- $F^{(n)} = w_b \ln(b^{(n)}) + w_c \ln(1/c^{(n)})$ , donde  $w_b + w_c = 1$

Ejemplo 2:

- Supóngase que la red #1 ofrece el doble ancho de banda  $b$  y el doble retardo  $d$  que la red #2, y que los pesos para el ancho de banda y para el retardo son iguales ( $w = 0.5$ ). Entonces, el usuario piensa que los *fitness* de ambas redes son iguales, por lo que ambas redes son iguales de buenas:
- $F^{(1)} = w \ln(1/b^{(1)}) + w \ln(c^{(1)}) = w \ln(c1/b1)$
- $F^{(2)} = w \ln(1/b^{(2)}) + w \ln(c^{(2)}) = w \ln(c2/b2)$
- $b^{(1)} = 2 b^{(2)}$ ,  $c^{(1)} = 2 c^{(2)}$ , entonces podemos comprobar que  $F^{(1)} = F^{(2)}$ .

### 6.2.5 Parámetros QoS

Dependiendo del tipo de parámetro de calidad de servicio que se vaya a considerar, se tendrá que utilizar la ecuación (30) o la ecuación (31) a la hora de aplicar la función de *fitness*.

En la Tabla 12 se muestran algunos parámetros relacionados con la calidad del servicio [Chiasserini 2004] que van a ser utilizados en las distintas instancias experimentales que se abordan en el problema de optimización.

Hay muchos otros parámetros relacionados con la calidad del servicio no contemplados en dicha tabla, y que no han sido considerados en esta tesis al no estar incluidos en las instancias experimentales utilizadas. Por ejemplo: RSS (*received signal strength*, fuerza de la señal recibida), el área de cobertura, la distancia entre el terminal móvil y la estación base, el consumo de potencia, etc.

### 6.2.6 Vertical Handoff en redes de sensores inalámbricos

Las Redes de Sensores Inalámbricas (WSNs en sus siglas en inglés, de *Wireless Sensor Networks*) constituyen una tecnología cuyo uso se ha incrementado sustancialmente en los últimos años [Yick 2008]. El éxito de esta tecnología y su versatilidad han favorecido la aparición de aplicaciones en diversas áreas, tanto en el ámbito civil (control industrial, monitorización medioambiental, agricultura de precisión, sistemas de protección anti-incendio, salud y dependencia, etc.) como en el militar (operaciones de rescate, supervivencia, etc.).

Por otro lado, los avances producidos en el ámbito de las redes y comunicaciones inalámbricas son constantemente aplicados a las redes de sensores inalámbricos, para propiciar la aparición de nuevas formas de implementar servicios y aplicaciones en esta tecnología. Así por ejemplo, bajo la convergencia de estas áreas (WSN, redes y comunicaciones), la movilidad emerge como un paradigma de gran importancia que proporciona muchas posibilidades, como por ejemplo las redes de sensores móviles.

### **6.2.6.1 Movilidad en redes de sensores inalámbricos**

Hay muchos casos de WSNs donde la movilidad es una característica importante que hay que tener en cuenta. Por ejemplo, la posición de sensores móviles en una WSN debe ser determinada a veces con exactitud para el correcto funcionamiento de sistemas de eventos donde la trazabilidad y localización deben ser monitorizados constantemente [Shah 2010][Hu 2004].

Otra aplicación de los sensores móviles es para ayudar en el despliegue inicial de una WSN [WangG 2005], ya que los sensores móviles pueden moverse a localizaciones en las que puedan encontrar suficiente grado de conectividad, asegurando así la construcción de una red óptima. También es interesante la aplicación de los sensores móviles es la detección de intrusos [Keung 2010], donde los sensores móviles pueden mejorar la cobertura de la barrera de protección contra intrusos móviles.

Por tanto, la movilidad aplicada a las WSNs es un campo de investigación interesante que abarca muchos frentes, siendo abordado tanto desde el punto de vista académico como el industrial. Así por ejemplo, en el NASA Glenn Research Center [NASA 2011] se investiga en plataformas móviles para posicionamiento de sensores, exploración espacial, monitorización medioambiental, tecnología de redes móviles aplicables a plataformas de sensores móviles, etc.

Finalmente, un aspecto muy importante para ser investigado es la conectividad de sensores móviles con redes heterogéneas de sensores inalámbricos, teniendo en cuenta la calidad de servicio: este es el área que se aborda en esta tesis.

### **6.2.6.2 Conectividad de WSNs heterogéneas basada en calidad de servicio**

Esta tesis estudia el caso complejo del proceso VH basado en calidad de servicio de un sensor inalámbrico móvil que se desplaza a lo largo de redes de sensores inalámbricas heterogéneas (es decir, varias redes con diferentes tecnologías que proporcionan, por tanto, distintos valores de los parámetros de calidad de servicio).

Se puede explicar mejor el marco de este estudio mediante el siguiente ejemplo.

La Figura 37 muestra un caso donde están presentes tres redes de sensores inalámbricas distintas. Cada red tiene:

- Un nodo colector (al que llega la información de toda la red).
- Un conjunto de routers como puntos de acceso a la red (RA).
- un conjunto de routers como pasarelas (RP) entre las coberturas proporcionadas por los RA, de forma que cualquier sensor localizado dentro del alcance de los RA pueda hacer llegar su información hasta el correspondiente nodo colector.

Los dos tipos de routers, RA y RP, pueden incluso ser también sensores. Cada red tiene configurada, a través de estos routers, el camino de datos para hacer llegar la información de los sensores al correspondiente colector, sensores que pueden estar libremente localizados.

Pues bien, un sensor móvil que se desplace a lo largo de estas tres redes puede establecer su comunicación en ocasiones con más de un router de acceso, decidiendo en cada momento a cuál conectarse (y por tanto, a qué red de las tres, a la que pertenezca ese router) en función de los perfiles definidos de QoS. En este caso, se produce el proceso de decisión VH al tratarse de redes que pueden ser de distinta tecnología, con distintos valores de los parámetros QoS.

Supóngase un sensor inalámbrico móvil desplazándose a través de este escenario (Figura 38). En su camino, algunos routers quedan a su alcance, donde éstos pueden pertenecer a distintas redes. En esta situación, el sensor inalámbrico móvil debe decidir en todo momento cuál es el mejor router al cuál conectarse, esto es, cuál es el router que pertenece a la red que le proporciona la mejor QoS.

Como se puede observar en la Figura 38, el sensor móvil puede encontrar más de un router en determinados momentos en su trayectoria, donde los routers pueden ser de la misma red (*Horizontal Handoff*) o de distintas redes (*Vertical Handoff*).

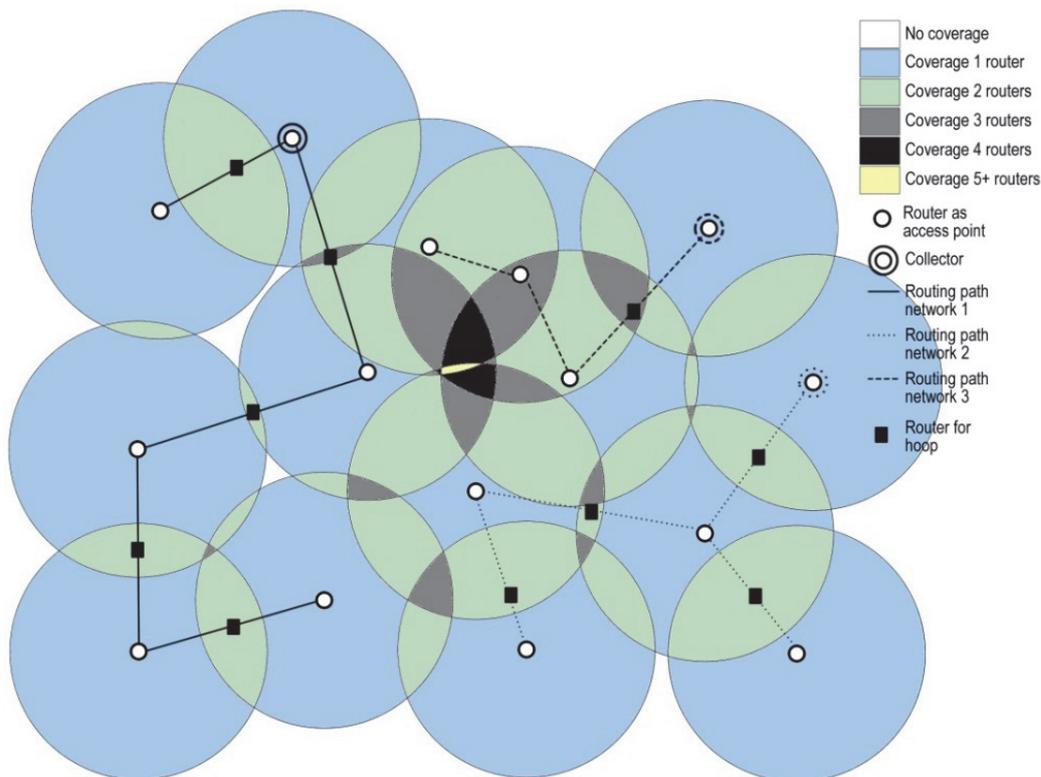


Figura 37. Red de sensores inalámbricos heterogénea con tres redes basadas en routers de diferente tecnología, proporcionando distintos valores para los parámetros QoS, y preparada para dar soporte a sensores inalámbricos móviles.

Parámetro QoS	Unidad	Definición	Para minimizar <i>fitness</i> hay que:	Si aumenta, la calidad de la red:
Productividad, ancho de banda ( <i>throughput, bandwidth</i> )	Mb/s	Cantidad de datos que se puede enviar a través de una conexión de red por unidad de tiempo.	Maximizar	Mejora
Retardo ( <i>delay</i> )	ms	Tiempo que emplea la señal para atravesar el canal de transmisión.	Minimizar	Empeora
Tiempo de respuesta, latencia ( <i>response time, latency</i> )	ms	Tiempo que tarda la información en llegar al receptor desde que es enviada por el emisor.	Minimizar	Empeora
Congestión de red ( <i>jitte</i> )	ms	Variación en el tiempo en la llegada de los paquetes.	Minimizar	Empeora
BER ( <i>Bit Error Ratio</i> )	dB	Tasa de bits con error respecto a los bits recibidos en una transmisión.	Minimizar	Empeora
<i>Burst error</i>	const.	Mide la secuencia de símbolos contiguos erróneos recibidos durante una transmisión de datos.	Minimizar	Empeora
Retransmisiones de paquetes promedio ( <i>average packet retransmissions</i> ).	const.	Número promedio de retransmisiones de paquetes de datos en una comunicación.	Minimizar	Empeora
Tasa de pérdida de paquetes ( <i>packet loss ratio</i> )	%	Tasa de pérdida de paquetes de datos en una comunicación.	Minimizar	Empeora
Nivel de seguridad ( <i>security level</i> )	const.	Nivel de seguridad (0 representa una red no segura, y 5 muy segura).	Maximizar	Mejora
Coste económico ( <i>cost</i> )	eur/kB	Coste económico de la transmisión de datos o utilización de la red.	Minimizar	Empeora
RSSI ( <i>Received Signal Strength Indication</i> )	dBm	Indicador de fuerza de señal de recepción, mide la potencia presente en la señal de radio.	Maximizar	Mejora
Cobertura de red ( <i>network coverage</i> )	m	Área cubierta por la red inalámbrica.	Maximizar	Mejora
Robustez ( <i>reliability</i> )	const.	Nivel de robustez de la red. Puede ser establecido según qué forma.	Maximizar	Mejora
Potencia del terminal ( <i>power requirement</i> )	W	Mide el nivel de consumo eléctrico del terminal móvil	Minimizar	Empeora
Velocidad del terminal	m/s	Velocidad a la que el terminal móvil se desplaza en el instante en que se evalúa el VH.	Maximizar	Mejora

Tabla 12. Parámetros QoS más habituales. Algunos de estos parámetros deben ser maximizados o minimizados para poder minimizar el *fitness* (o maximizar la función de mérito); maximizar el parámetro implica mejorar el *fitness* (a mayor p, menor *fitness*, tal como se ve en la ecuación (31)), mientras que minimizar el parámetro implica el empeoramiento del *fitness* (a mayor p, mayor *fitness*, tal como se ve en la ecuación (30)).

En el caso de conexión VH, se pueden producir distintas situaciones. Por ejemplo, para las posiciones #3 y #25 no es posible conectarse con ningún router, porque el terminal móvil está dentro de áreas sin cobertura y los valores emitidos por el sensor móvil (por ejemplo, la temperatura adquirida en su entorno) no pueden ser recogidos por ningún nodo colector.

El proceso VH sucede, por ejemplo, en la posición #8, donde se pueden alcanzar las tres redes, porque hay hasta 5 routers pertenecientes a esas redes en el área de cobertura del sensor móvil; en este caso, el sensor móvil debe decidir a qué red conectarse, de acuerdo a los valores proporcionados por los parámetros de calidad de servicio. Por otro lado, el proceso HH sucede, por ejemplo, en la posición #16, donde pueden ser alcanzados dos routers distintos pertenecientes a la misma red; en este caso, el sensor continúa conectado a la misma red.

La Figura 39 muestra el escenario completo, combinando la Figura 38 y la Figura 39, para así tener una visión completa de los posibles procesos VH y HH.

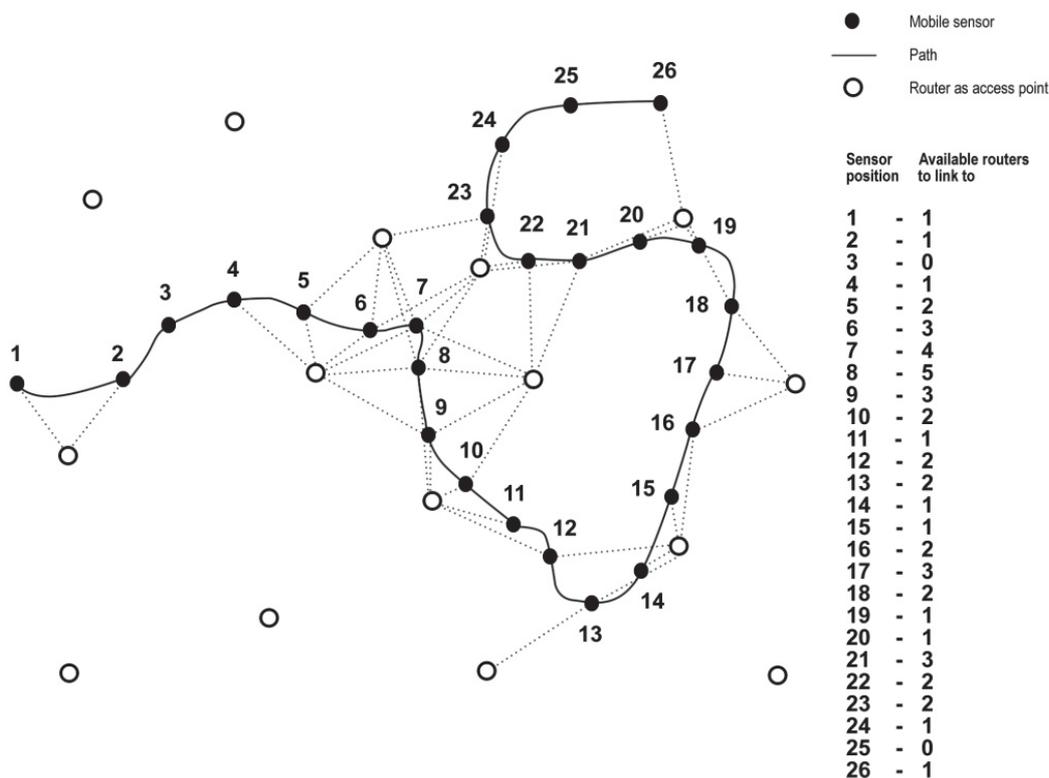


Figura 38. Un sensor inalámbrico móvil configura un camino donde puede ser alcanzado por la conexión de varios routers de distintas redes heterogéneas. Este esquema se basa en el escenario de la Figura 37. La primera columna indica la posición del sensor móvil en la trayectoria; la segunda informa de cuántos routers disponibles están a su alcance.

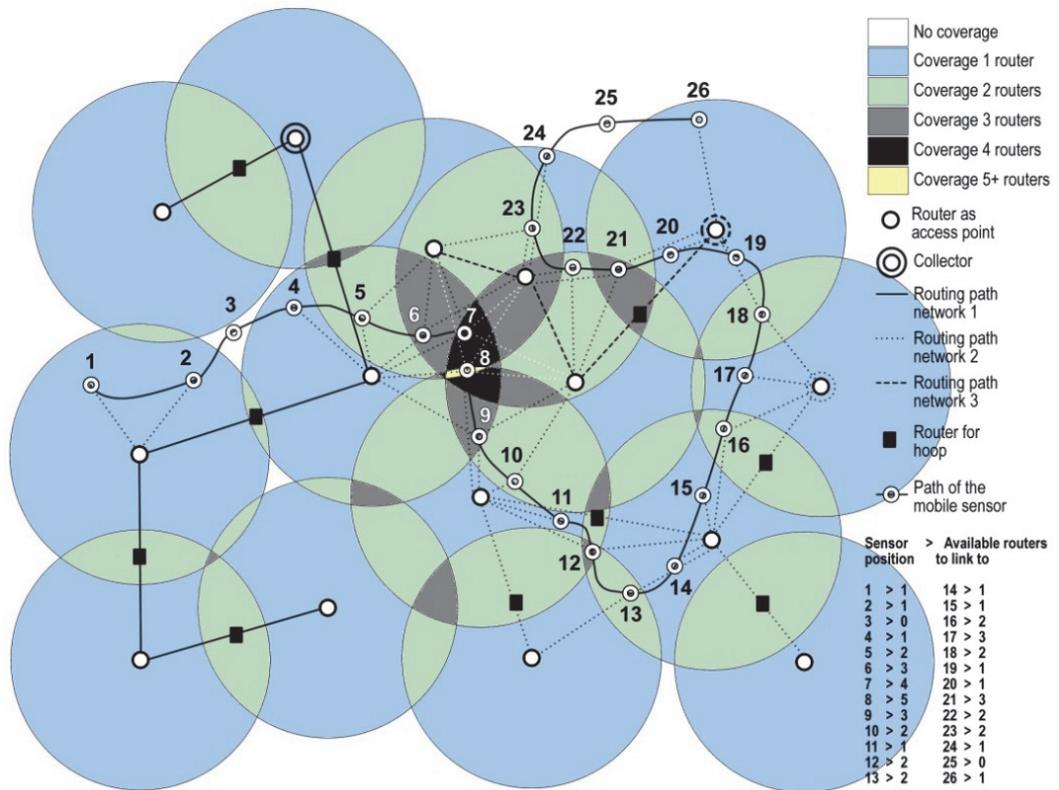


Figura 39. Escenario completo para una red de sensores inalámbricos heterogénea donde pueden darse distintos procesos para VH/HH para el sensor móvil.

## 6.3 Problemática de optimización

El presente trabajo de tesis se enmarca en escenarios donde un terminal móvil necesita conectividad ininterrumpida a la red, siempre con la mejor calidad de servicio para las preferencias definidas por el usuario. Esto implica que, cuando el terminal descubre varias redes inalámbricas como puntos de acceso a las que puede conectarse, debe determinar cuál de ellas ofrece la mejor calidad de servicio utilizando la función de *fitness* anteriormente descrita.

Este problema de optimización no es trivial, computacionalmente es muy costoso, debido principalmente al ajuste de los pesos, cuyas combinaciones son muy numerosas según la precisión requerida o el número de parámetros de calidad.

### 6.3.1 *Ajuste de los pesos*

El ajuste óptimo de los pesos para la determinación de la red que proporcione la mejor solución para la decisión VH es una tarea que requiere de heurísticas de optimización, dado el enorme tamaño del espacio de soluciones posible (tamaño que aumenta cuanto mayor sea la precisión que se requiera para los pesos). Entre otras posibilidades, se ha utilizado la técnica *Analytic Hierarchy Process (AHP)* [Song 2005]. También se pueden asignar directamente valores a los pesos por parte del usuario cuando una llamada es inicializada y sintonizada dinámicamente de acuerdo al QoS percibido, pero esta sería una técnica subjetiva y hace perder tiempo al usuario [Song 2008].

Las preferencias del usuario pueden definirse en el terminal móvil mediante unos perfiles donde cada peso puede tomar unos valores dentro de unos rangos predefinidos, siempre y cuando respeten la restricción del problema (27). Estos perfiles pueden cambiarse en función de las condiciones del terminal o de los intereses del usuario.

Así por ejemplo, en el ajuste de los pesos se podría incrementar el peso del consumo de potencia cuando el terminal está con poca batería. Esto requiere, en cualquier caso, volver a aplicar una optimización de los pesos asignados a cada parámetro QoS para determinar la red óptima a la cual conectarse.

### 6.3.2 *Características del problema y definiciones*

Para elaborar un planteamiento adecuado que aborde el problema de optimización, han de considerarse las siguientes características del mismo:

- **Sistemas dinámicos.** El terminal que ha de conectarse con una red inalámbrica es un **terminal móvil** (dispositivo a bordo de un automóvil, un helicóptero, un tren, un avión), por lo que los parámetros de la red pueden cambiar rápidamente en el tiempo (Figura 40). Esto exige encontrar soluciones en muy poco tiempo, cercanas al **tiempo real**, para que los recursos del microprocesador dedicados a la decisión VH no menoscaben en el normal funcionamiento del dispositivo.

- **Sistemas integrados.** Los requisitos de alto nivel de integración de los dispositivos móviles (bajo consumo y pequeño tamaño) descartan la posibilidad de ejecutar complejas heurísticas que consuman grandes cantidades de recursos de memoria y que tengan elevados tiempos de ejecución, ya que los microprocesadores empotrados suelen tener un rendimiento menor que el de los microprocesadores de varios núcleos incluidos en los ordenadores personales, por ejemplo. Esto limita la elección o implementación de los algoritmos de optimización que resuelvan el problema de la decisión VH basada en QoS.
- **Combinación.** Es un conjunto o vector de pesos asignados a los parámetros de calidad de servicio, donde cada peso tiene un valor comprendido entre 0 y 1.

$$(32) \quad \text{combinación} = \{w_0, w_1, \dots, w_{NQ-1}\}, \text{ donde } NQ \text{ es el número de parámetros QoS y donde } 0 \leq w_i \leq 1$$

- **Solución.** Una solución es una combinación de pesos de los parámetros QoS que cumple la restricción dada en la ecuación (27). También llamaremos a la solución como “combinación válida”.

$$(33) \quad \text{solución} = \{w_0, w_1, \dots, w_{NQ-1}\}, \text{ donde } NQ \text{ es el número de parámetros QoS y donde } \sum_{i=0}^{NQ-1} w_i = 1$$

Es importante remarcar que cada solución se puede aplicar a distintas redes, ofreciendo distintos valores de *fitness*, ya que cada red tiene sus propios valores de los parámetros QoS.

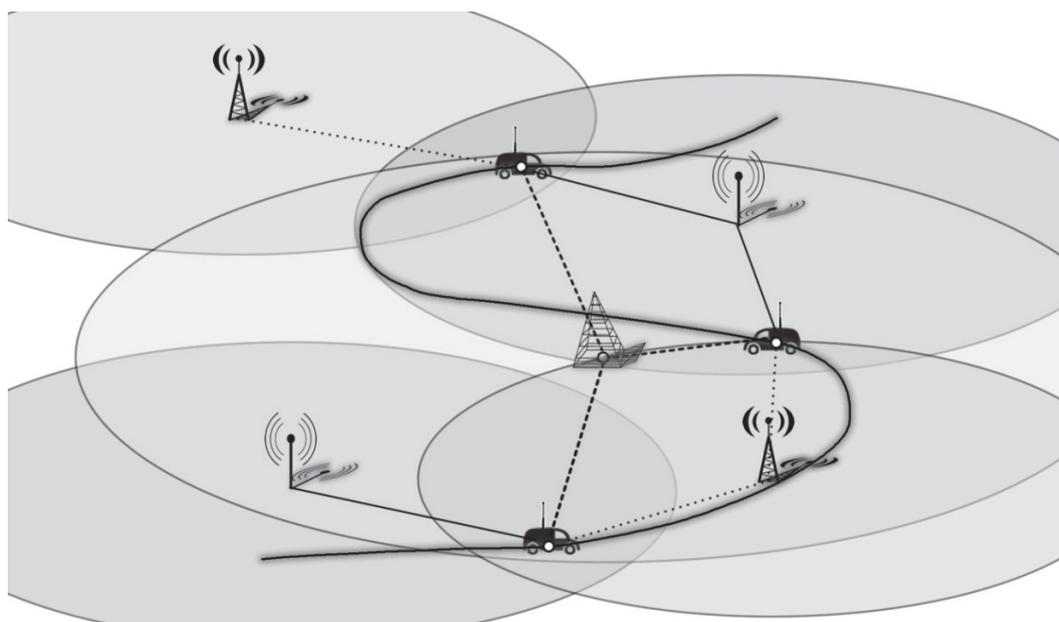


Figura 40. Terminal móvil que, en su desplazamiento dinámico, descubre varias redes inalámbricas con distintas calidades de servicio.

- **Solución óptima de cada red.** De entre todas las posibles soluciones para una misma red, la mejor u óptima es aquella que tiene el menor valor de *fitness*.
- **Solución óptima del problema.** De entre todas las soluciones óptimas de cada red, la que tenga el menor *fitness* es la solución óptima del problema, determinando así cuál es la red óptima para la fase de decisión VH.
- **Búsqueda de la solución óptima.** El problema de optimización consiste en encontrar una solución óptima de entre todas las posibles. Esto podría hacerse mediante técnicas exactas (como búsquedas directas) o heurísticas (como algoritmos evolutivos), según sea el coste computacional involucrado en la búsqueda (que depende de la cantidad de posibles soluciones).

## 6.4 Instancias experimentales

Para validar las heurísticas desarrolladas, determinar el coste computacional del problema, etc., se han utilizado tres instancias experimentales.

### 6.4.1 *Instancia 1*

Esta instancia [Song 2005] contempla un área con tres redes WLAN (AP1, AP2, AP3) y una red UMTS (BS) (Figura 41). Las distancias desde AP2 a AP1 y AP3 son 400m y 100m respectivamente.

Esta instancia ha sido utilizada en una simulación donde un usuario móvil, que transfiere un archivo de datos, se mueve en sentido contrario a las agujas del reloj a 1 m/s desde BS a AP1, AP2, AP3 y luego vuelve al punto de inicio.

Esta instancia experimental tiene un gran interés para esta tesis en cuanto a que contempla un elevado número de parámetros de calidad de servicio, tal como se puede ver en la Tabla 13. Esto permite introducir un elevado coste computacional a los algoritmos de optimización.

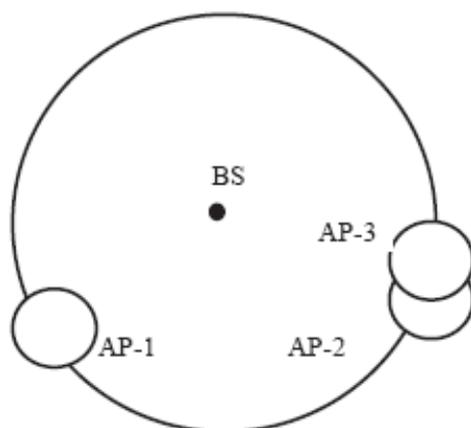


Figura 41. Instancia experimental número 1. Consiste en 4 redes inalámbricas que proporcionan hasta 10 parámetros de calidad del servicio

Red	#0 - BS	#1 - AP1	#2 - AP2	#3 - AP3	
<b>Tipo</b>	UMTS	WLAN	WLAN	WLAN	
<b>Cobertura</b>	1000 m	200 m	200 m	200 m	
<b>Parámetros QoS</b>					
$\alpha$ (Mb/s) - throughput	1.7	25	20	25	
$\beta$ - timeliness	$\zeta$ (ms) - delay	19	30	45	50
	$\eta$ (ms) - response time	9	30	28	30
	$\theta$ (ms) - jitter	6	10	10	10
$\gamma$ - reliability	$\lambda$ (dB) - BER	0.001	0.00001	0.00001	0.000001
	$\mu$ - burst error	0.5	0.2	0.25	0.2
	$\nu$ - average number of retransmissions per packet	0.4	0.2	0.3	0.2
	$\sigma$ - packet loss ratio	0.07	0.05	0.04	0.04
$\delta$ (level) - security	8	7	6.5	6	
$\epsilon$ (per kbyte) - cost	0.9	0.1	0.2	0.5	

Tabla 13. Características de las redes para la instancia #1.

Valores de las variables que utilizarán las heurísticas desarrolladas:

- Número máximo de redes: InstanceMaxNN = 4.
- Número máximo de parámetros QoS: InstanceMaxNQ = 10.
- Parámetros QoS:
  - QoSPar[0] = "Throughput, Bandwidth" (productividad, ancho de banda).
  - QoSPar[1] = "Delay" (retardo).
  - QoSPar[2] = "Response time, Latency" (tiempo de respuesta, latencia).
  - QoSPar[3] = "Jitter" (variación en el tiempo en llegar los paquetes).
  - QoSPar[4] = "BER" (tasa de error de bit).
  - QoSPar[5] = "Burst error".
  - QoSPar[6] = "A average number of retransmissions per packet"
  - QoSPar[7] = "Packet loss ratio" (tasa de pérdida de paquetes).
  - QoSPar[8] = "security (level)" (nivel de seguridad de la red).
  - QoSPar[9] = "Cost (per kbyte)" (coste económico del servicio por kbyte).
- Valores de los parámetros QoS para cada red (QoS[red][parámetro]):

```

QoS [0] [0]=1.7;
QoS [1] [0]=25;
QoS [2] [0]=20;
QoS [3] [0]=25;
QoS [0] [1]=19;
QoS [1] [1]=30;
QoS [2] [1]=45;
QoS [3] [1]=50;
QoS [0] [2]=9;
QoS [1] [2]=30;
QoS [2] [2]=28;
QoS [3] [2]=30;
QoS [0] [3]=6;

```

```
QoS[1][3]=10;  
QoS[2][3]=10;  
QoS[3][3]=10;  
QoS[0][4]=0.001;  
QoS[1][4]=0.00001;  
QoS[2][4]=0.00001;  
QoS[3][4]=0.000001;  
QoS[0][5]=0.5;  
QoS[1][5]=0.2;  
QoS[2][5]=0.25;  
QoS[3][5]=0.2;  
QoS[0][6]=0.4;  
QoS[1][6]=0.2;  
QoS[2][6]=0.3;  
QoS[3][6]=0.2;  
QoS[0][7]=0.07;  
QoS[1][7]=0.05;  
QoS[2][7]=0.04;  
QoS[3][7]=0.04;  
QoS[0][8]=8;  
QoS[1][8]=7;  
QoS[2][8]=6.5;  
QoS[3][8]=6;  
QoS[0][9]=0.9;  
QoS[1][9]=0.1;  
QoS[2][9]=0.2;  
QoS[3][9]=0.5;
```

## 6.4.2 Instancia 2

Esta instancia [Lassoued 2008] permite considerar dos tipos de servicios de comunicaciones, para aplicaciones de conversación y de *streaming* (los parámetros QoS más adecuados para cada una de estas aplicaciones, además de su importancia relativa, se definen en [QoS 2008]).

En esta instancia el terminal móvil se desplaza en un entorno heterogéneo (Figura 42) formado por 6 redes inalámbricas cuyas características se definen en la Tabla 14. El nivel de seguridad ofrecido por las diferentes redes de acceso varía entre 0 (que representa una red no segura) y 5 (que representa una red de alta seguridad).

Valores de las variables que utilizarán las heurísticas desarrolladas:

- Número máximo de redes: InstanceMaxNN = 6.
- Número máximo de parámetros QoS: InstanceMaxNQ = 5.
- Parámetros QoS:
  - QoSPar[0] = "Bandwidth (kbps)".
  - QoSPar[1] = "BER".
  - QoSPar[2] = "Delay (ms)".
  - QoSPar[3] = "Security (level)".
  - QoSPar[4] = "Monetary cost (eur/MB)".

Valores de los parámetros QoS para cada red (QoS[red][parámetro]):

```
QoS[0][0]=5100;  
QoS[1][0]=5100;  
QoS[2][0]=256;  
QoS[3][0]=5100;  
QoS[4][0]=5100;  
QoS[5][0]=384;  
QoS[0][1]=0.01;  
QoS[1][1]=0.01;  
QoS[2][1]=0.01;  
QoS[3][1]=0.01;  
QoS[4][1]=0.01;  
QoS[5][1]=0.03;  
QoS[0][2]=70;  
QoS[1][2]=65;  
QoS[2][2]=85;  
QoS[3][2]=75;  
QoS[4][2]=55;  
QoS[5][2]=80;  
QoS[0][3]=2;  
QoS[1][3]=1;  
QoS[2][3]=3;  
QoS[3][3]=3;  
QoS[4][3]=3;  
QoS[5][3]=5;  
QoS[0][4]=0.2;  
QoS[1][4]=0.2;  
QoS[2][4]=0.3;  
QoS[3][4]=0.2;  
QoS[4][4]=0.2;  
QoS[5][4]=0.2;
```

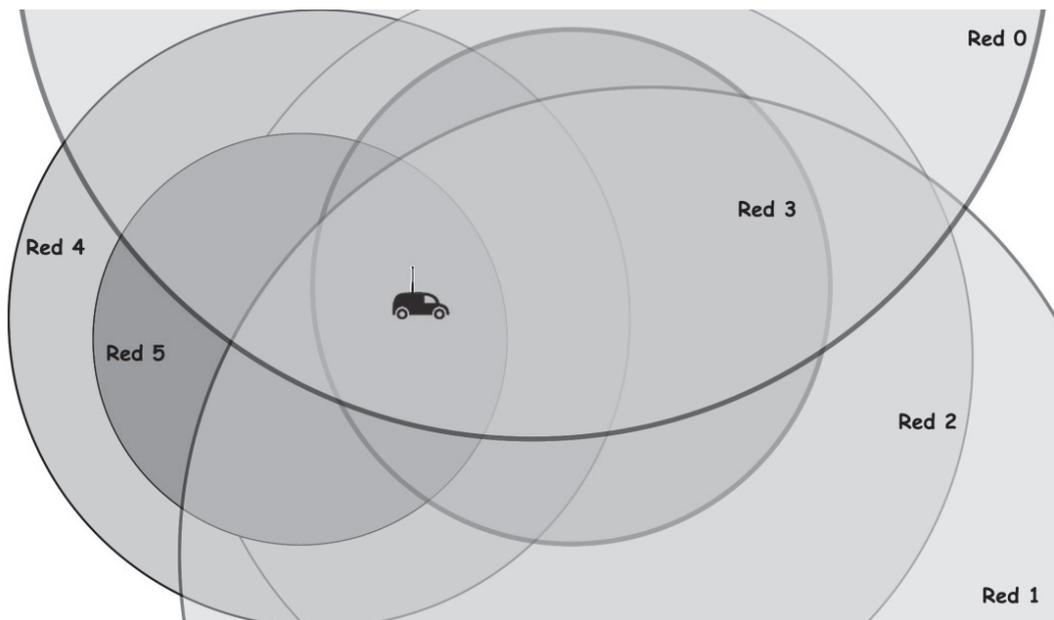


Figura 42. Instancia experimental número 2. Consiste en 6 redes inalámbricas que proporcionan hasta 4 parámetros de calidad del servicio.

### 6.4.3 Instancia 3

En [Nkansah 2007] se detalla un escenario que simula un día típico en la vida de un usuario de redes inalámbricas (Mr. Alex). Este usuario se desplaza en un escenario en el que están siempre disponibles tres redes inalámbricas de distinta tecnología para teléfonos móviles, GPRS\_1, UMTS\_1 y UMTS\_2, parcialmente presentes dos redes inalámbricas de área local, WLAN\_P\_1 y WLAN\_P\_2, y una sexta red, solo disponible en la oficina de Mr. Alex, WLAN\_O.

En el escenario simulado se contemplan dos casos de utilización de las redes:

- **Caso A.** Durante el almuerzo, Mr. Alex ha comenzado a descargar algunos archivos multimedia utilizando la red UMTS\_1; en su desplazamiento, descubre las dos redes WLAN\_1 y WLAN\_2, y desea utilizar la red más barata que proporcione alta transferencia de datos para completar la descarga de los archivos. En este caso, a la tasa de transferencia de datos se le otorga (mediante su correspondiente peso) la máxima importancia sobre el resto de parámetros QoS, seguida del coste económico como factor más importante.
- **Caso B.** Durante la tarde, Mr. Alex participa en una videoconferencia, en la misma trayectoria del caso A. Aquí, la tasa de transferencia de datos es de máxima importancia, seguida por el coste económico y la latencia de red, por este orden, sobre el resto de parámetros QoS.

Red	Tecnología	Ancho de banda	BER	Retardo	Seguridad	Coste (eur/MB)
0	IEEE802.11b	5,100 kbps	0.01	70 ms	2	0.2
1	IEEE802.11b	5,100 kbps	0.01	65 ms	1	0.2
2	WiMax	256 kbps	0.01	85 ms	3	0.3
3	IEEE802.11b	5,100 kbps	0.01	75 ms	3	0.2
4	IEEE802.11b	5,100 kbps	0.01	55 ms	3	0.2
5	UMTS	384 kbps	0.03	80 ms	5	0.2

Tabla 14. Características de las redes para la instancia #2. La medida del ancho de banda de las distintas redes se obtiene a partir de fuentes referidas por [Lassoued 2008]. Así, para redes IEEE802.11b, el valor está sacado de [Atheros 2001] teniendo en cuenta escenarios realistas, ya que el ancho de banda depende de varios factores, entre ellos la distancia emisor-receptor. La medida del ancho de banda para las redes UMTS es aportada directamente en [Lassoued 2008], mientras que la de las redes WiMax (IEEE 802.16) está sacada de [Betancur 2006].

Estos dos casos conforman dos servicios de usuario móvil de redes inalámbricas, que serán descritos y cuantificados en el siguiente apartado, 6.4.4.

La utilización de esta instancia, tal como se describe en [Nkansah 2007], implica evaluar la calidad de las redes WLAN\_1 y WLAN\_2 para establecer un proceso de VH desde UMTS\_1. Para ello, se aporta una función de fitness, llamada "*multiple attribute wireless network selection function*" (**WNSF**). Esta función de fitness es utilizada para la fase de decisión VH (que responde a una formulación de problema tipo MCDM, donde se define una serie de atributos o parámetros QoS), y es evaluada para cada red.

Es importante hacer notar que esta función de fitness ha sido formulada de forma que la red óptima es la de valor más alto de WNSF, en contraste con la función de fitness  $F$  formulada en (29), que debe ser minimizada para obtener la red óptima. En otras palabras, los autores de este trabajo maximizan WNSF para indicar una mayor calidad de la red.

Otra diferencia de esta instancia en relación con las anteriores dos, es que los autores no aportan los valores reales de los parámetros QoS de cada red, sino lo que ellos llaman "valores de pertenencia",  $x_i$  (*membership values*).

Por ejemplo, el parámetro RSSI (*signal strength*) puede tener valores que oscilan entre -78dBm y -66dBm, o el parámetro ancho de banda (*data rate*) puede oscilar entre 0 7 56Mbs, pero estos valores no se aportan para las 3 redes considerados. En su lugar, se aportan los valores  $x_i$  que se muestran en la Tabla 15, y que corresponden con la formulación determinada de la función WNSF, formulación distinta a la considerada en esta tesis según la ecuación (29).

Para poder utilizar esta instancia en los algoritmos de optimización desarrollados en esta tesis, de forma análoga a cómo se han utilizado las otras dos instancias, y que consideran la función de fitness según (29), se han calculado los valores de los parámetros QoS ( $p$  en dicha ecuación) de las redes a partir de  $x_i$  de forma que:

$$(34) \quad \ln(p') = x \rightarrow p' = e^x \rightarrow \begin{cases} p = e^x & \text{si aumentar } p \text{ mejora la calidad de red (ancho de banda)} \\ p = \frac{1}{e^x} & \text{si aumentar } p \text{ empeora la calidad de red (coste económico)} \end{cases}$$

Parámetro QoS:	S	D	A	L	R	E	P	V	C
Red	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	x(7)	x(8)	x(9)
0 (UMTS_1)	0.9	0.1	0.6	0.6	0.9	0.9	0.8	0.9	0.6
1 (WLAN_P_1)	0.9	0.7	0.2	0.8	0.8	0.65	0.6	0.01	0.9
2 (WLAN_P_2)	0.9	0.9	0.1	0.85	0.8	0.6	0.6	0.01	0.9

Red	p(1)	p(2)	p(3)	p(4)	p(5)	p(6)	p(7)	p(8)	p(9)
0 (UMTS_1)	2.460	1.105	1.822	0.549	2.460	2.460	0.449	2.460	0.549
1 (WLAN_P_1)	2.460	2.014	1.221	0.449	2.226	1.916	0.549	1.010	0.407
2 (WLAN_P_2)	2.460	2.460	1.105	0.427	2.226	1.822	0.549	1.010	0.407

Tabla 15. Características de las redes para la instancia #3.

Así, los valores de  $p$ , ó parámetros QoS de las redes, se muestran en la misma Tabla 15. Se ha podido validar, a partir de una determinada combinación de pesos y valor de WNSF aportados como datos en [Nkansah 2007], que el valor así obtenido por  $F$  según (29) coincide con el aportado de WNSF en dicho artículo.

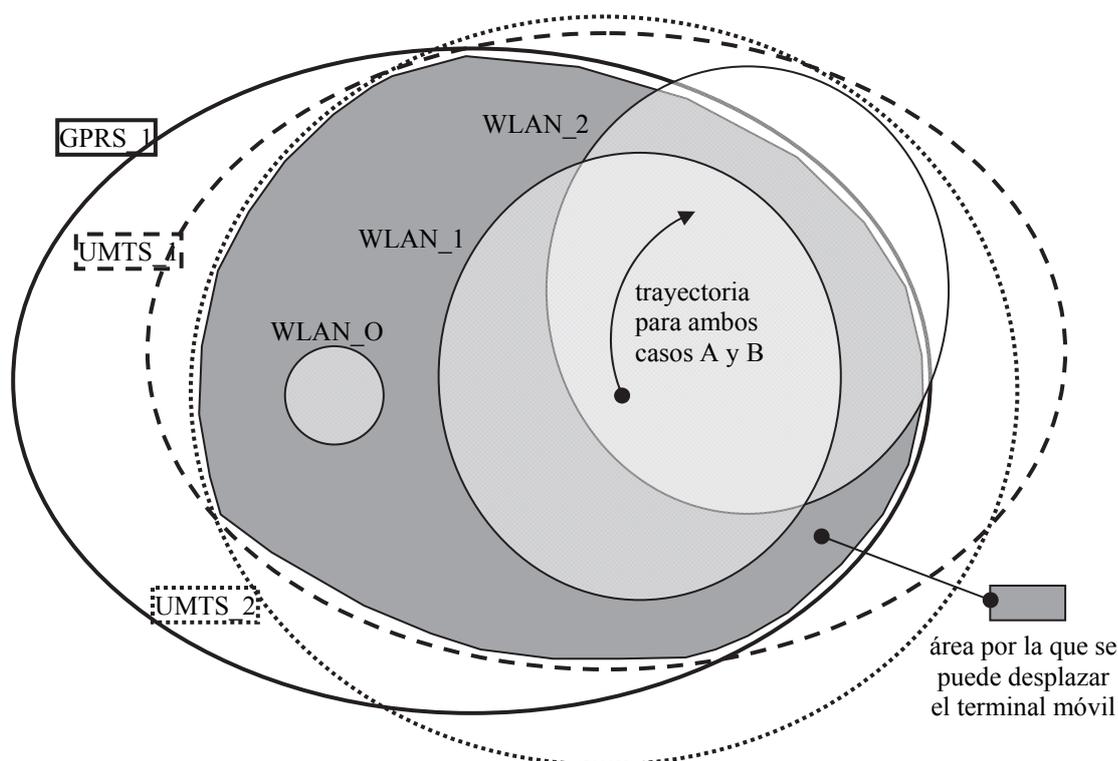


Figura 43. Escenario correspondiente a la instancia #3, donde existen 6 redes inalámbricas. El terminal móvil se puede desplazar en un área donde hay siempre presentes tres redes de telefonía móvil: dos inalámbricas de área local presentes de forma parcial y una de área local aislada. Los casos A y B de la instancia corresponden a un VH que se produce desde la red inalámbrica UMTS\_1 a dos posibles redes inalámbricas de área local, WLAN\_1 y WLAN\_2, teniendo en cuenta dos servicios de comunicaciones distintos.

Los valores de las variables que utilizarán las heurísticas desarrolladas son:

- Número máximo de redes: InstanceMaxNN = 3.
- Número máximo de parámetros QoS: InstanceMaxNQ = 9.
- Parámetros QoS:
  - QoSPar[0] = "RSSI (**S**)".
  - QoSPar[1] = "Data rate (**D**)".
  - QoSPar[2] = "Network coverage (**A**)".
  - QoSPar[3] = "Network latency (**L**)".
  - QoSPar[4] = "Reliability (**R**)".
  - QoSPar[5] = "Security (**E**)".
  - QoSPar[6] = "Power requirement (**P**)".
  - QoSPar[7] = "Mobile velocity (**V**)".
  - QoSPar[8] = "Service cost (**C**)".

Valores de los parámetros QoS para cada red (QoS[red][parámetro]):

```
QoS[0][0]=2.459603111;
QoS[1][0]=2.459603111;
QoS[2][0]=2.459603111;
QoS[0][1]=1.105170918;
QoS[1][1]=2.013752707;
QoS[2][1]=2.459603111;
QoS[0][2]=1.8221188;
QoS[1][2]=1.221402758;
QoS[2][2]=1.105170918;
QoS[0][3]=0.548811636;
QoS[1][3]=0.449328964;
QoS[2][3]=0.427414932;
QoS[0][4]=2.459603111;
QoS[1][4]=2.225540928;
QoS[2][4]=2.225540928;
QoS[0][5]=2.459603111;
QoS[1][5]=1.915540829;
QoS[2][5]=1.8221188;
QoS[0][6]=0.449328964;
QoS[1][6]=0.548811636;
QoS[2][6]=0.548811636;
QoS[0][7]=2.459603111;
QoS[1][7]=1.010050167;
QoS[2][7]=1.010050167;
QoS[0][8]=0.548811636;
QoS[1][8]=0.40656966;
QoS[2][8]=0.40656966;
```

#### **6.4.4 Perfiles con las preferencias del usuario**

Para las dos primeras instancias experimentales, Instancia 1 e Instancia 2, se han definido tres posibles perfiles de usuario comunes en el uso del terminal móvil:

- Perfil #1 (Instancia 1 e Instancia 2): **Perfil genérico.**

Es el perfil más genérico posible, donde el usuario no especifica ninguna restricción, rango o preferencia por uno u otro parámetro. En este perfil se han establecido los siguientes rangos para los parámetros QoS:

- Todos los parámetros QoS pueden tener cualquier peso que esté comprendido entre 0 y 1.
- Perfil #2 (Instancia 1 e Instancia 2): **Perfil de conversación.**
- En este perfil cobra una mayor importancia el retardo y el coste económico frente a otros factores de calidad del servicio, ya que una conversación debe ser procesada en tiempo real y no ha de ser muy cara, ya que no suelen ser de muy corta duración. En este perfil se han establecido los siguientes rangos para los parámetros QoS:
- Retardo: Peso comprendido entre 0.5 y 0.7.

- Perfil #3 (Instancia 1 e Instancia 2): **Perfil de streaming.**

Este es un perfil típico para aplicaciones de transmisión multimedia (visualización de videos, por ejemplo). En este perfil cobra importancia el ancho de banda, para que se puedan transmitir la mayor cantidad posible de bytes por unidad de tiempo. En este perfil se han establecido los siguientes rangos para los parámetros QoS:

- Ancho de banda: Peso comprendido entre 0.5 y 0.7.
- Retardo: Peso comprendido entre 0.1 y 0.3.

Para la tercera instancia experimental se han definido dos perfiles de usuario, correspondientes a sus casos A y B:

- Perfil #4 (Instancia 3): **Perfil de descarga de archivos.**

Perfil utilizado en el caso A de la instancia 3, donde el parámetro más importante es la tasa de transferencia de datos (ancho de banda), seguido por el coste económico. En este perfil se han establecido dos casos o sub-perfiles con los siguientes rangos para los parámetros QoS:

Sub-perfil 4.A:

- Ancho de banda: Peso comprendido entre 0.6 y 0.8.
- Coste del servicio: Peso comprendido entre 0.1 y 0.2.

Sub-perfil 4.B:

- Ancho de banda: Peso comprendido entre 0.5 y 0.7.
- Coste del servicio: Peso comprendido entre 0.2 y 0.4.

- Perfil #5 (Instancia 3): **Perfil de videoconferencia.**

Perfil utilizado en el caso B de la instancia 3, donde el parámetro más importante es la tasa de transferencia de datos (ancho de banda), seguido por el coste económico y la latencia de red, por este orden. En este perfil se han establecido los siguientes rangos para los parámetros QoS:

- Ancho de banda: Peso comprendido entre 0.4 y 0.6.
- Coste del servicio: Peso comprendido entre 0.2 y 0.3.
- Latencia de red: Peso comprendido entre 0.1 y 0.2.

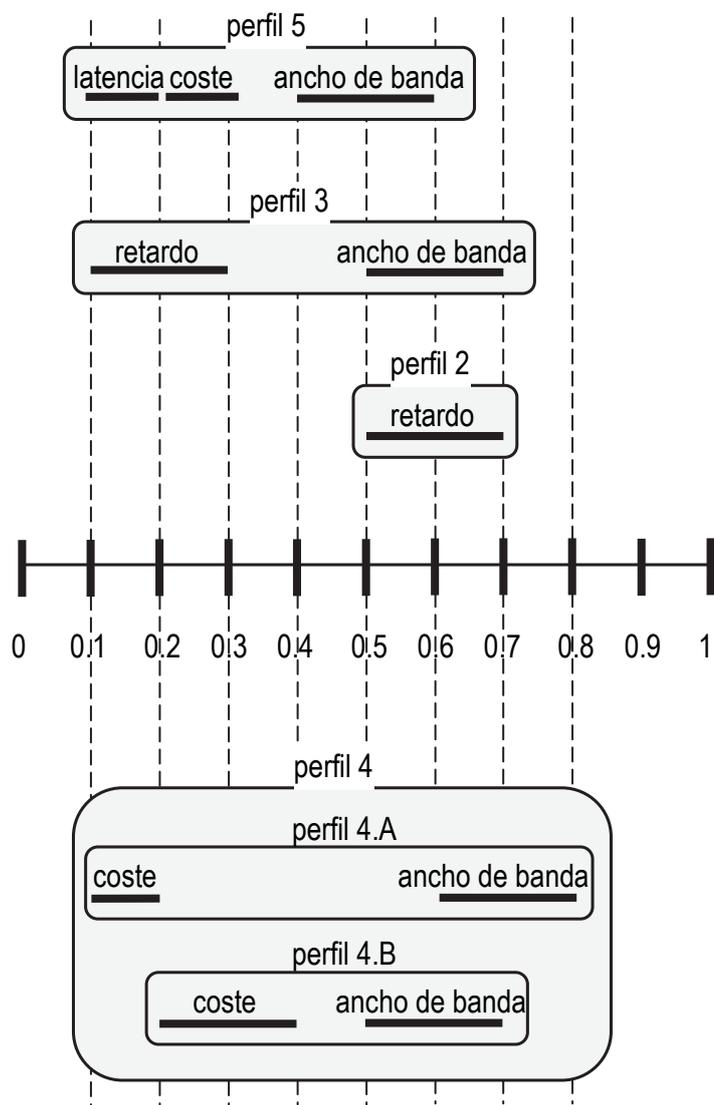


Figura 44. Rangos de pesos para parámetros QoS según los perfiles de usuario.

## 6.5 Planteamientos para el problema de optimización

### 6.5.1 *Determinación del coste computacional*

Para determinar la complejidad del problema en cuanto a su coste computacional, así como para validar las heurísticas desarrolladas, se han realizado unas experimentaciones previas que calculan el número de soluciones posibles en función del número de redes y del número de parámetros de calidad del servicio.

Para efectuar estas experimentaciones se han necesitado dos cosas: unas instancias experimentales del problema y un algoritmo de búsqueda directa pero no exhaustiva. Las instancias ya han sido descritas anteriormente, y el algoritmo se detalla en la siguiente sección.

En este apartado se adelantan algunos de los resultados obtenidos por el algoritmo de búsqueda directa (SEFI) con el propósito de demostrar el elevado coste computacional que tiene el problema de optimización, en cuanto a hacer notar la necesidad de implementar heurísticas que proporcionen buenas soluciones en un tiempo mínimo para no interferir con las operaciones habituales del terminal.

En la Figura 45 se puede observar el tiempo de computación que el algoritmo SEFI consume para distintas precisiones en la búsqueda de soluciones, y también para distinto número de parámetros QoS. El algoritmo consume tiempo básicamente en tres cosas:

- Generación de las combinaciones.
- Determinación de las combinaciones válidas (que cumplen la restricción).
- Cálculo del *fitness* de las combinaciones válidas.

La Figura 46 muestra el número de combinaciones generadas por SEFI y las que cumplen la restricción (combinaciones válidas), para distintos valores de la precisión en la búsqueda de soluciones y para distinto número de parámetros QoS.

Del análisis de ambas figuras se pueden concluir dos cosas:

- Los tiempos de computación aumentan mucho cuantos más parámetros QoS se consideren, cuantas más redes inalámbricas existan y cuanto mayor precisión de la solución se requiera, ya que estos tres factores elevan drásticamente el número de combinaciones que se pueden generar y que deben ser evaluadas.
- La precisión en la generación de combinaciones (y por tanto, en la búsqueda de soluciones) es el principal factor que incrementa el espacio de búsqueda a tamaños enormes, asemejándose el problema de optimización a un problema NP-duro. Se demostrará que elevar esta precisión puede suponer, en algún caso, escoger una red distinta a la que conectarse en la fase de decisión VH, por lo que trabajar con precisiones bajas para computar más rápido no es una buena política.

Por todo ello, se apunta a la necesidad de abordar el problema con algoritmos evolutivos o bien de búsqueda exacta, aunque no exhaustiva (que no evalúan la totalidad de las soluciones posibles, ya que computacionalmente es inasumible).

Como se explicará más adelante, los algoritmos de búsqueda exacta no exhaustiva (no evolutivos) tienen la ventaja de cubrir de manera más uniforme todo el espacio de soluciones, pero al buscar las soluciones a intervalos regulares pueden desecharse posibles soluciones óptimas.

Por el contrario, los algoritmos evolutivos pueden encontrar más fácilmente las soluciones óptimas, pero son más complejos de implementar y programar para este problema de optimización, puesto que la restricción del problema requiere que los individuos generados tengan que ser modificados para cumplir dicha restricción. De esta forma, se corre el peligro de que dicha adaptación de sus genes provoque dejar zonas del espacio de soluciones sin explorar.

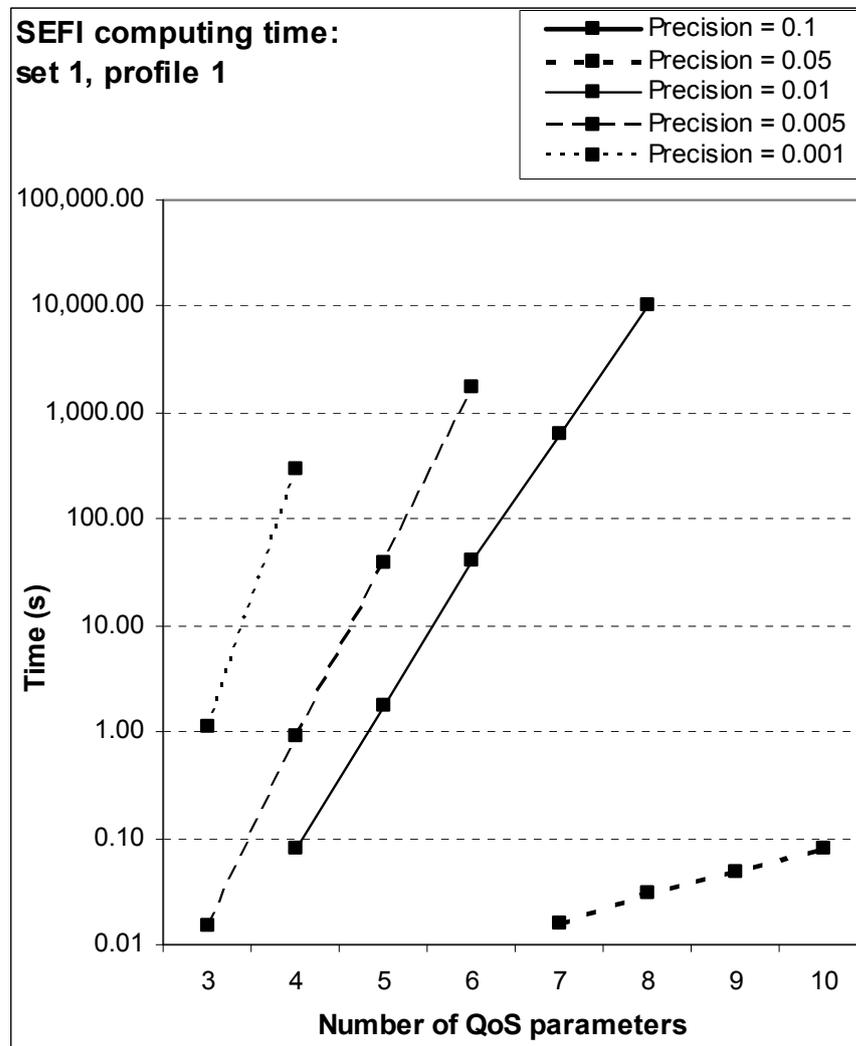


Figura 45. Tiempos de computación del algoritmo SEFI para la instancia experimental #1, teniendo en cuenta distintos valores de la precisión en la búsqueda de soluciones para distinto número de parámetros QoS. A mayor precisión, mayor número de combinaciones generadas. El tiempo de computación se emplea tanto para generar las combinaciones como para calcular el *fitness* de cada una de ellas.

### 6.5.2 ¿Mono-objetivo o multi-objetivo?

Existen dos posibilidades para enfocar el problema de optimización, de forma que pueda ser abordado, por ejemplo, por un **Algoritmo Evolutivo (AE)**:

- Optimización **mono-objetivo**. Este enfoque trata de minimizar el *fitness* global, es decir, ajustar todos los pesos  $w_i$  correspondientes a todos los parámetros QoS.
- Optimización **multi-objetivo**. Este enfoque trata de minimizar el *fitness* global más el *fitness* de un parámetro particular de QoS que se considera aparte. En [Lassoued 2008] se plantea esta posibilidad: “*UA uses two weights to solve the multi-objectives problem: ( $\gamma$ ) for the QoS and ( $1-\gamma$ ) for the monetary cost allow to balance the decision between the goal to reduce the cost and the goal to have the better QoS which are often contradictory*”.

Es decir, este enfoque considera ajustar los pesos  $w_i$  para minimizar por un lado el coste económico como un objetivo y minimizar el *fitness* global por otro lado, con la ventaja (para el problema de optimización) de que ambos objetivos son contradictorios.

La decisión adoptada en esta tesis para abordar el problema de optimización ha sido **considerar el enfoque mono-objetivo**, por las siguientes razones:

- Los algoritmos que abordan la optimización multi-objetivo suelen ser más costosos de implementar (consumiendo más recursos de memoria y teniendo mayores tiempos de ejecución), cosa que entra seriamente en conflicto con los intereses y aplicabilidad del problema: decisiones en tiempo real (velocidad de ejecución) e integración en microprocesadores empotrados de bajo consumo (algoritmos ligeros) para dispositivos móviles.
- En el enfoque mono-objetivo se ha encontrado un trabajo con el cual poder comparar resultados, mientras que no se ha localizado ningún resultado contrastable de un enfoque multi-objetivo que siga el planteamiento general del problema de optimización VH tal como se expone en este capítulo. Esta consideración es importante, ya que la validación de las heurísticas propuestas en esta tesis solo puede venir de su comparación con las propuestas por otros autores que aborden el mismo problema de optimización.

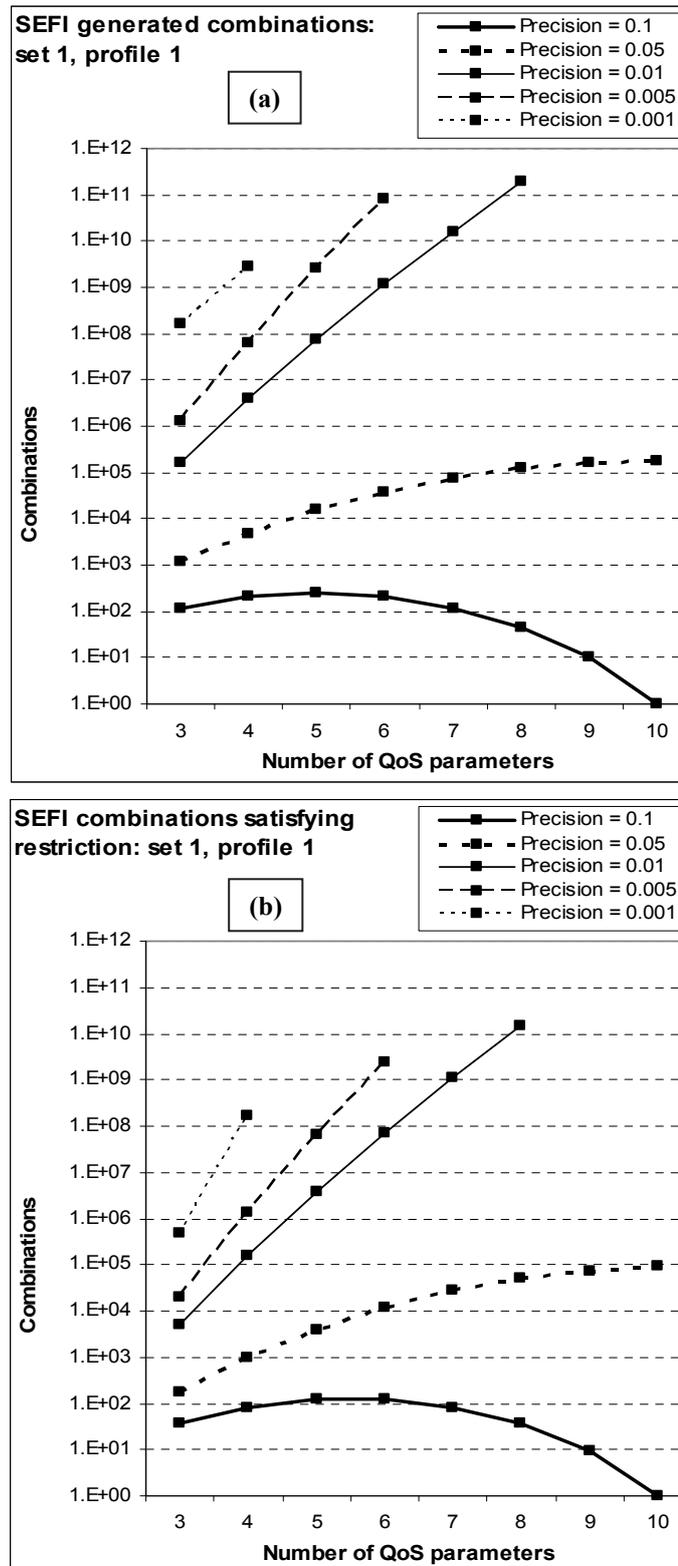


Figura 46. Número de combinaciones generadas por SEFI (a) y número de éstas que cumplen la restricción del problema de optimización (b), para la instancia experimental #1, teniendo en cuenta distintos valores de la precisión en la búsqueda de soluciones para distinto número de parámetros QoS.

### 6.5.3 Algoritmo de optimización

Para decidir qué tipo de algoritmo o heurística desarrollar para abordar el enfoque de optimización mono-objetivo, se han tenido en cuenta los siguientes aspectos:

- Es necesario desarrollar un algoritmo ligero y rápido, capaz de encontrar una solución óptima en tiempos pequeños, idealmente próximos al tiempo real.
- Hay que idear un algoritmo que pueda explorar soluciones que se encuentren, de forma representativa o uniformemente distribuidas, por todo el espacio de soluciones. De esta forma el algoritmo evita dejar zonas del espacio de soluciones sin explorar, ya que podrían ignorarse áreas donde estén las mejores soluciones.
- Sería aconsejable programar los algoritmos en lenguaje C para que puedan ejecutarse en microprocesadores empujados, bien sean de tipo “duro” (ya implementados o cableados) o “blando” (configurables, como por ejemplo Microblaze en una FPGA).
- La restricción impuesta a las posibles soluciones (que la suma de los pesos sea igual a uno) es lo suficientemente exigente como para complicar la evolución de individuos (soluciones) de una población en un algoritmo evolutivo. De hecho, esta restricción condiciona enormemente la implementación de los algoritmos y complica mucho su programación.

En un algoritmo evolutivo, operadores como el cruce y la mutación deberían tener, en este caso, algún tipo de acción adicional que permita que los individuos resultantes de aplicar los operadores sigan satisfaciendo dicha restricción. Esto podría hacer que los individuos resultantes, una vez aplicadas dichas correcciones finales para satisfacer la restricción, se alejasen de zonas del espacio de soluciones, permitiendo la posibilidad de que pudiera haber alguna zona permanentemente sin ser explorada (se contempla evitar esto mediante políticas de mutación).

Todas estas características han motivado el desarrollo de un algoritmo de búsqueda directa (cubriendo de manera uniforme todo el espacio de soluciones) no exhaustiva (para una precisión dada, debido al enorme tamaño del espacio de soluciones), como primera aproximación para resolver el problema de optimización: el algoritmo SEFI.

## 6.6 Algoritmo SEFI

En esta tesis se ha desarrollado un algoritmo de **búsqueda directa pero no exhaustiva**. Con esto se quiere decir que el algoritmo realiza una búsqueda directa de soluciones pero en el espacio de soluciones existente para una precisión dada. A mayor precisión, mayor es el tamaño del espacio de soluciones y, por tanto, más tiempo tardará la búsqueda directa en completarse.

Se ha puesto el nombre de **SEFI** al algoritmo para identificar los términos “Weight Combinations **SE**arch by **FI**xed **I**ntervals”.

### 6.6.1 *Características*

El algoritmo se define por las siguientes características:

- En relación con el análisis de las combinaciones:
  - Genera todas las combinaciones posibles para una precisión ( $h$ ) dada.
  - Analiza todas las combinaciones generadas para determinar las que satisfacen la restricción del problema (combinaciones o soluciones válidas).
  - Calcula el *fitness* de todas las soluciones válidas en búsqueda del óptimo global.
- En relación con las entradas, el algoritmo pide:
  - El número de redes (NN), establecido en la fase de descubrimiento VH.
  - El máximo número de parámetros QoS (NQ) que proporcionan las redes.
  - La precisión de la búsqueda. Se trata de establecer el intervalo para generar los valores entre 0 y 1 ( $h$ ) que se asignarán a los pesos de los parámetros QoS, con la restricción de que  $0.5 > h > 10^{-9}$  (esta restricción se debe a que se han considerado en  $C$  variables enteras sin signo que no pueden exceder de la precisión 4,294,967,295).
- En relación con las salidas, SEFI proporciona para cada uno de los perfiles, y para cada una de las redes:
  - Tiempo de computación (segundos).
  - Combinaciones generadas.
  - Combinaciones generadas que satisfacen la restricción.
  - Combinación que proporciona el mejor valor de *fitness*, así como dicho valor.
  - Red inalámbrica que posee la combinación con el mejor valor de *fitness*, así como dicha combinación y dicho valor de *fitness*. Esta red será la escogida para la fase de ejecución VH.

## 6.6.2 Implementación

El algoritmo SEFI genera todas las posibles combinaciones teniendo en cuenta los límites impuestos por los perfiles o preferencias de usuario (Figura 47). A cada combinación se le aplica la comprobación de la restricción y, en caso de ser una solución válida, se calcula su *fitness* para cada red.

La clave para conseguir una búsqueda eficiente y que no descarte ninguna combinación está en la **recursividad** de una parte del algoritmo, que se llama a sí misma hasta alcanzar los límites máximos de búsqueda.

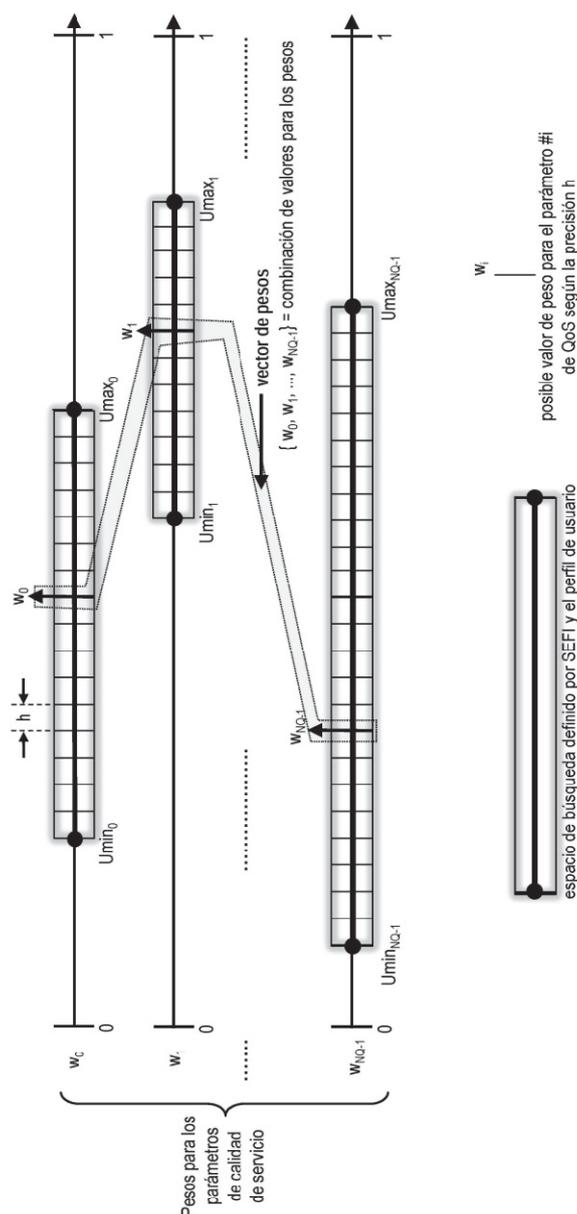


Figura 47. SEFI genera todas las posibles combinaciones (vectores de pesos) según una precisión dada  $h$ . Se comprueba para cada combinación que se cumpla la restricción y, en caso afirmativo, se calcula el *fitness* de esa solución para cada red. Las búsquedas estarán siempre acotadas a los límites de los pesos impuestos por los perfiles de usuario.

El algoritmo ha sido escrito completamente en C, utilizándose las típicas librerías estándar de ANSI C, con el propósito de asegurar la portabilidad del código desarrollado para poder ser ejecutado por cualquier microprocesador empujado en la mayoría de los sistemas operativos.

### 6.6.3 Resultados experimentales

En este apartado se muestran algunos resultados experimentales para la comprobación del coste computacional del algoritmo SEFI.

#### 6.6.3.1 Balance precisión/parámetros de calidad para asegurar el tiempo real

La complejidad computacional del problema es proporcional al nivel de precisión exigido en la búsqueda de soluciones y al número de parámetros de calidad del servicio. Al tener la restricción de considerar aplicaciones para terminales móviles, han de buscarse tiempos de respuesta cercanos al tiempo real. Por esta razón, en la búsqueda y análisis de los resultados de este algoritmo **se ha considerado un umbral de 1 segundo como el tiempo máximo para obtener una solución**. Esto limita, en función del número de redes inalámbricas, tanto el nivel de precisión como el número de parámetros QoS que se pueden considerar.

De esta forma, como puede observarse en la Figura 48, para precisiones iguales o inferiores a 0,01, los tiempos de computación se elevan demasiado, por encima de este umbral definido para el tiempo real.

Así, el análisis de estos resultados sugiere que, para respetar el carácter de tiempos pequeños, y para cualquier instancia experimental, **debe restringirse el funcionamiento de SEFI a los límites** señalados en la Tabla 16.

En relación con los tiempos de computación que aquí se muestran, cabe mencionar que el algoritmo SEFI ha sido ejecutado en un microprocesador comercial de rendimiento medio, Intel Core2 E6750 2.6GHz. Si bien algunos terminales móviles pueden tener microprocesadores más lentos, las conclusiones generales que se aportan en este apartado pueden ser extrapolables a cualquier caso, aunque se deben ajustar los valores de la Tabla 16 al correspondiente entorno hardware. En una sección posterior se mostrará detalladamente un caso de implementación hardware basada en FPGA para ilustrar este hecho.

NQ	h
2	$h \geq 0.001$
3	$h \geq 0.005$
4	$h \geq 0.010$
5	$h \geq 0.025$
6	$h \geq 0.025$
7	$h \geq 0.050$

Tabla 16. Valores de precisión que aseguran, para un número determinado de parámetros QoS, que el tiempo de computación de SEFI permanece por debajo de 1 segundo, umbral definido para mantener el carácter de tiempo real al proceso de selección de redes VH en entornos dinámicos.

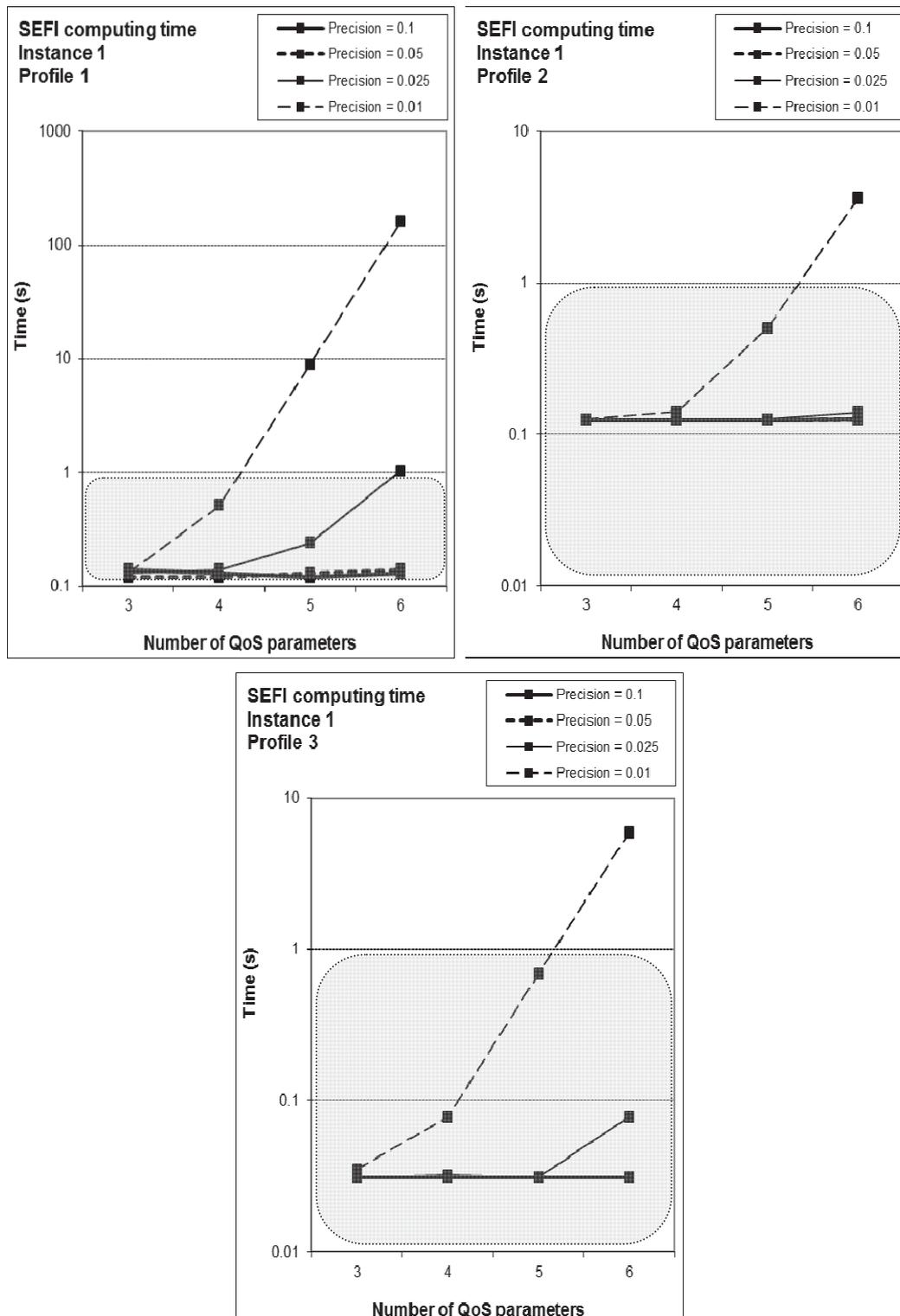


Figura 48. Tiempos de computación de SEFI para la instancia 1 (4 redes inalámbricas), en relación a hasta 6 parámetros QoS, para diversas precisiones de la búsqueda. Se ha señalado en cada gráfica, mediante una zona sombreada, los umbrales de tiempo real en 1 segundo.

### 6.6.3.2 Combinaciones generadas y soluciones del problema

La Figura 49 muestra el número de combinaciones generadas y evaluadas por el algoritmo SEFI (gráficas de la primera columna) y el número de estas combinaciones que satisfacen la restricción del problema de optimización, esto es, las posibles soluciones (gráficas de la segunda columna), frente al número de parámetros de calidad de servicio QoS, para la instancia experimental #1, y considerando distintos niveles de precisión en la generación de soluciones ( $h$ ).

Evidentemente, hay una relación directa entre el número de combinaciones generadas y evaluadas y el número de soluciones con el tiempo de computación. SEFI genera todas las combinaciones posibles para una precisión dada, y a todas ellas le aplica la evaluación o comprobación de que cumple la restricción dada en la ecuación (27); de entre aquellas que cumplen la restricción, denominadas soluciones, SEFI aplica la evaluación del *fitness* dada en la ecuación (29).

Por tanto, el tiempo de computación dependerá de todas estas evaluaciones y comprobaciones.

Se destacan como principales factores que inciden en el tiempo de computación, y por este orden, los siguientes:

- Evaluación del *fitness*.
- Evaluación de la restricción.
- Generación de una combinación.

Se pueden extraer a partir de la Figura 49 las siguientes consideraciones:

- El número de combinaciones y el número de soluciones aumenta con la precisión y con el número de parámetros de calidad de servicio. Esto es evidente ya que la precisión en la búsqueda aumenta considerablemente el espacio de posibles soluciones. Así por ejemplo, para un mismo número de parámetros de calidad ( $NQ=5$ ) y una alta precisión ( $h=0,01$ ), para el perfil #1 y la instancia #1, hay cerca de 100 millones de combinaciones generadas, de las cuales cerca del 7% son soluciones del problema de optimización.
- El número de combinaciones es mucho mayor que el número de soluciones. Esto es evidente, puesto que una solución es una combinación que cumple la restricción, por lo que hay pocas en relación al conjunto total de combinaciones.
- El número de soluciones tiende a disminuir cuanto mayor es el número de parámetros de calidad y, también, cuanto menor es la precisión de búsqueda. Este aspecto es sorprendente y llamativo. La posible explicación radica en la dificultad de encontrar combinaciones que satisfacen la restricción cuantos más elementos tiene el vector, y cuanto menor es la precisión de los mismos. Esta conclusión es empírica, hemos llegado a ella por análisis experimental, ya que desconocemos una formulación matemática que pueda explicarla.

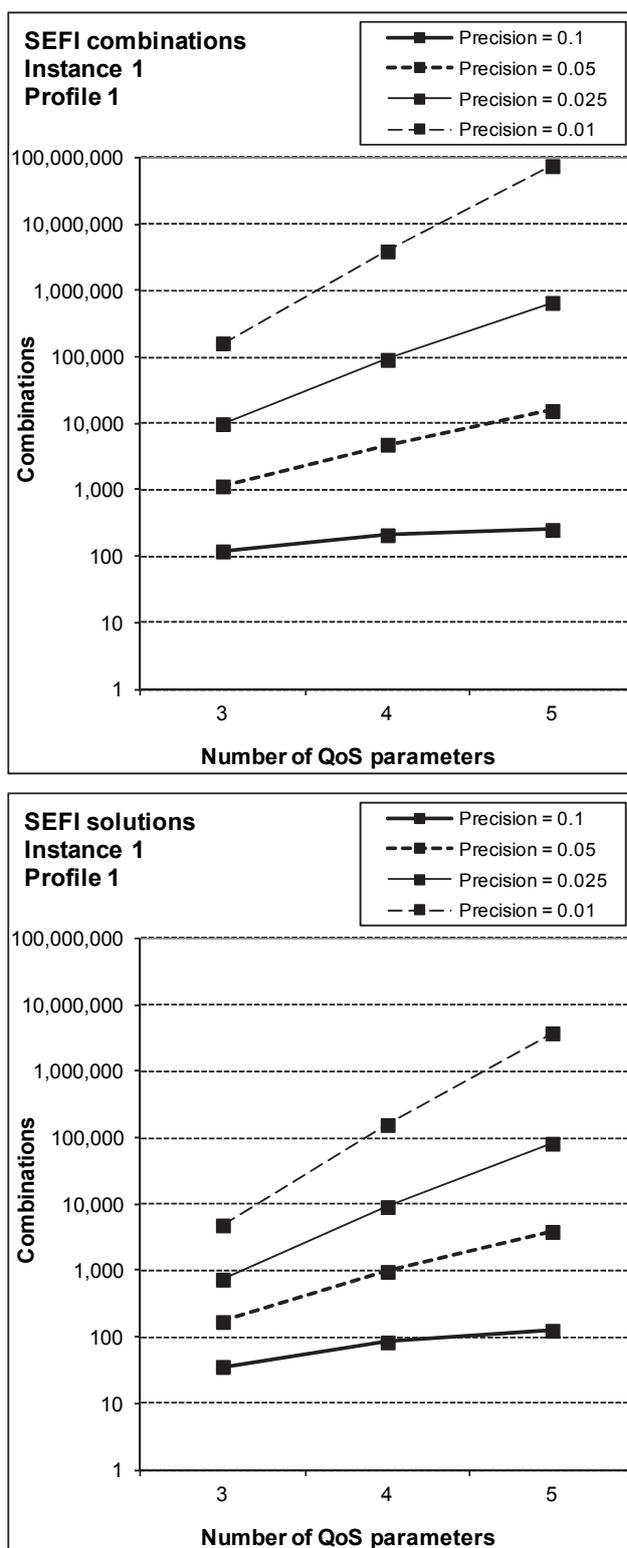


Figura 49. Número de combinaciones generadas y evaluadas por SEFI (gráfica superior de cada página) y número de éstas que satisfacen la restricción del problema (soluciones, gráfica inferior de cada página), frente al número de parámetros QoS, para la instancia experimental #1, y considerando distintos niveles de precisión en la búsqueda. Las gráficas de cada página corresponden a distintos perfiles de usuario.

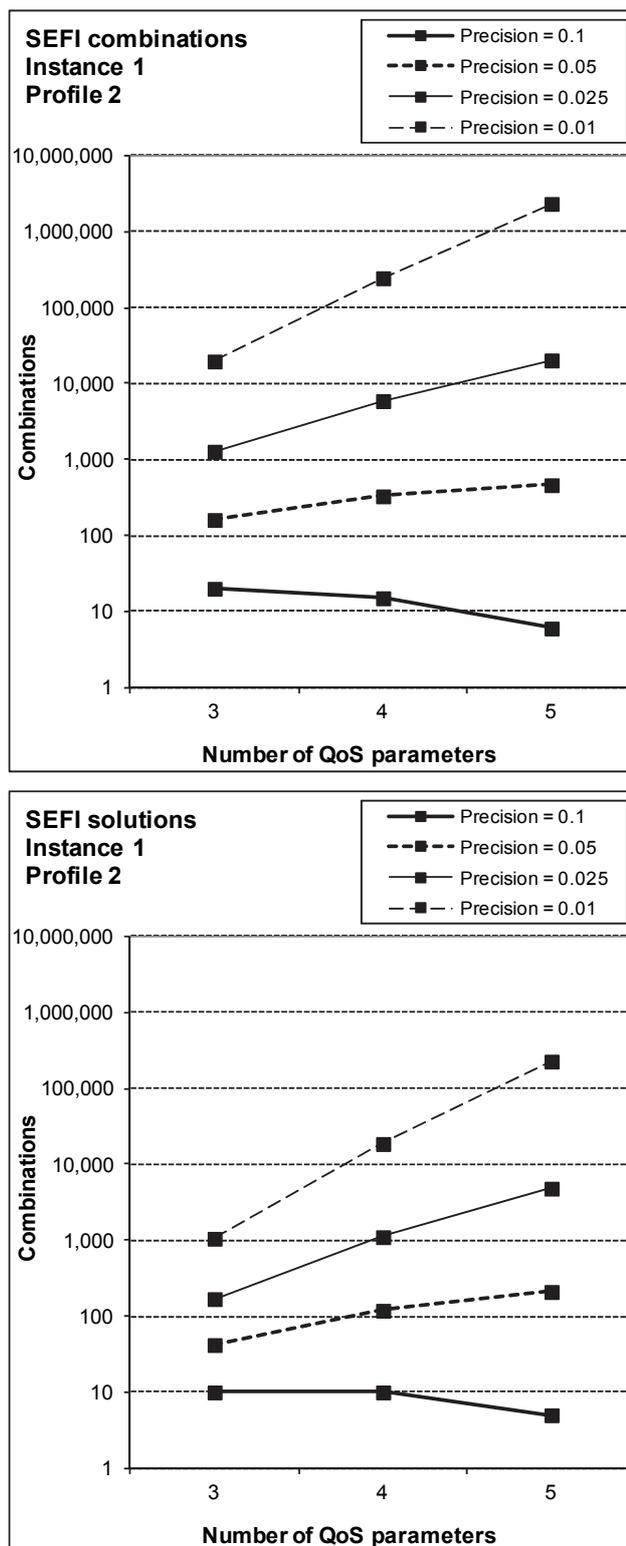


Figura 49. (continuación)

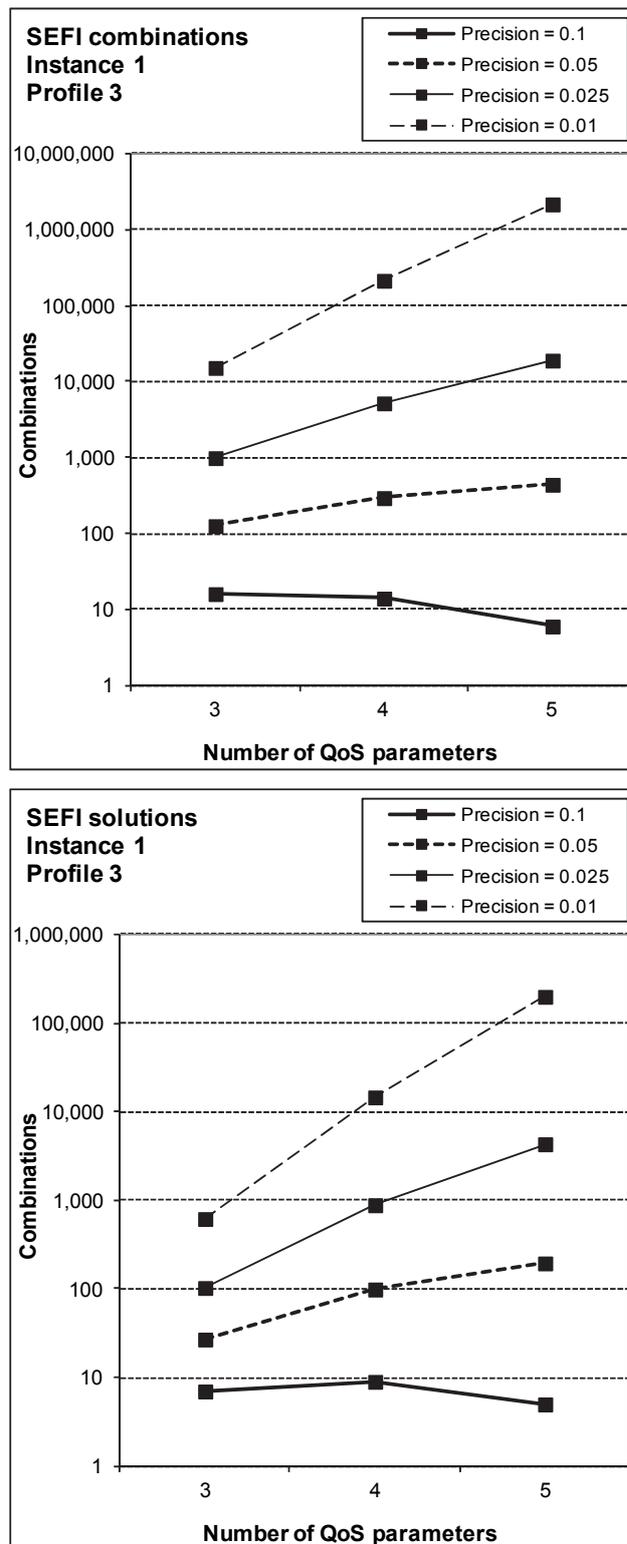


Figura 49. (continuación)

### 6.6.3.3 Obtención de una solución mejor

En la Figura 50 se puede observar el mejor *fitness* encontrado por SEFI (valor mínimo del *fitness* de todas las soluciones posibles encontradas) respecto al número de parámetros de calidad de servicio, para cada precisión. Los datos han sido obtenidos con la instancia #1, considerando los tres perfiles de usuario.

Se pueden observar, para todos los casos, dos hechos destacables:

- El *fitness* mejora con la precisión. Por tanto, a mayor precisión en la búsqueda, mejores soluciones encontradas.
- El *fitness* “empeora” cuantos más parámetros de calidad de servicio se consideren. Esto se debe a que el valor del *fitness* depende del número de parámetros QoS, tal como se refleja en la ecuación (29). Este hecho no debe llevar a una conclusión claramente errónea, como sería pensar que es mejor considerar menor número de parámetros QoS. La razón es porque, para cada valor de  $NQ$ , el problema de optimización es distinto, no es lo mismo considerar el problema añadiendo, por ejemplo, el nivel de seguridad de la red, que no hacerlo. En lo que hay que fijarse es que, para un valor concreto de  $NQ$ , siempre mejorará el *fitness* con la precisión.

La conclusión más significativa es que, mientras se pueda, hay que escoger una precisión de búsqueda mayor.

### 6.6.3.4 Necesidad de una heurística para la selección de una red óptima

Se ha comprobado que la selección de una red óptima puede depender tanto del número de parámetros de calidad de servicio considerados, como de la precisión en la búsqueda de un *fitness* mínimo.

En la Figura 51 se muestra un conjunto de gráficas con resultados relativos a la instancia experimental #1. Cada gráfica dibuja en el eje de ordenadas dos variables respecto al número (eje de abscisas) de parámetros de calidad de servicio considerado: el tiempo de computación de SEFI en encontrar la red óptima, y el identificador de esta red óptima (que puede ser 0, 1, 2 o 3, pues la instancia #1 consta de 4 redes). Además, cada gráfica está generada para un nivel de precisión determinado (valor  $h$ ).

De esta forma, se puede observar que:

- Al aumentar el número de parámetros QoS, se puede encontrar una red óptima distinta. Concretamente, a partir de 6 variables QoS, la red óptima es la #3, mientras que considerando de 2 a 5 variables QoS, la red óptima es la #1. Esto se constata para todos los perfiles considerados.
- En un caso concreto (perfil 2), la red óptima #3 no es descubierta hasta considerar 7 variables QoS, en lugar de 6 como en los otros casos. Sin embargo, al aumentar la precisión  $h$ , esta red óptima ya se puede encontrar con 6 variables QoS.

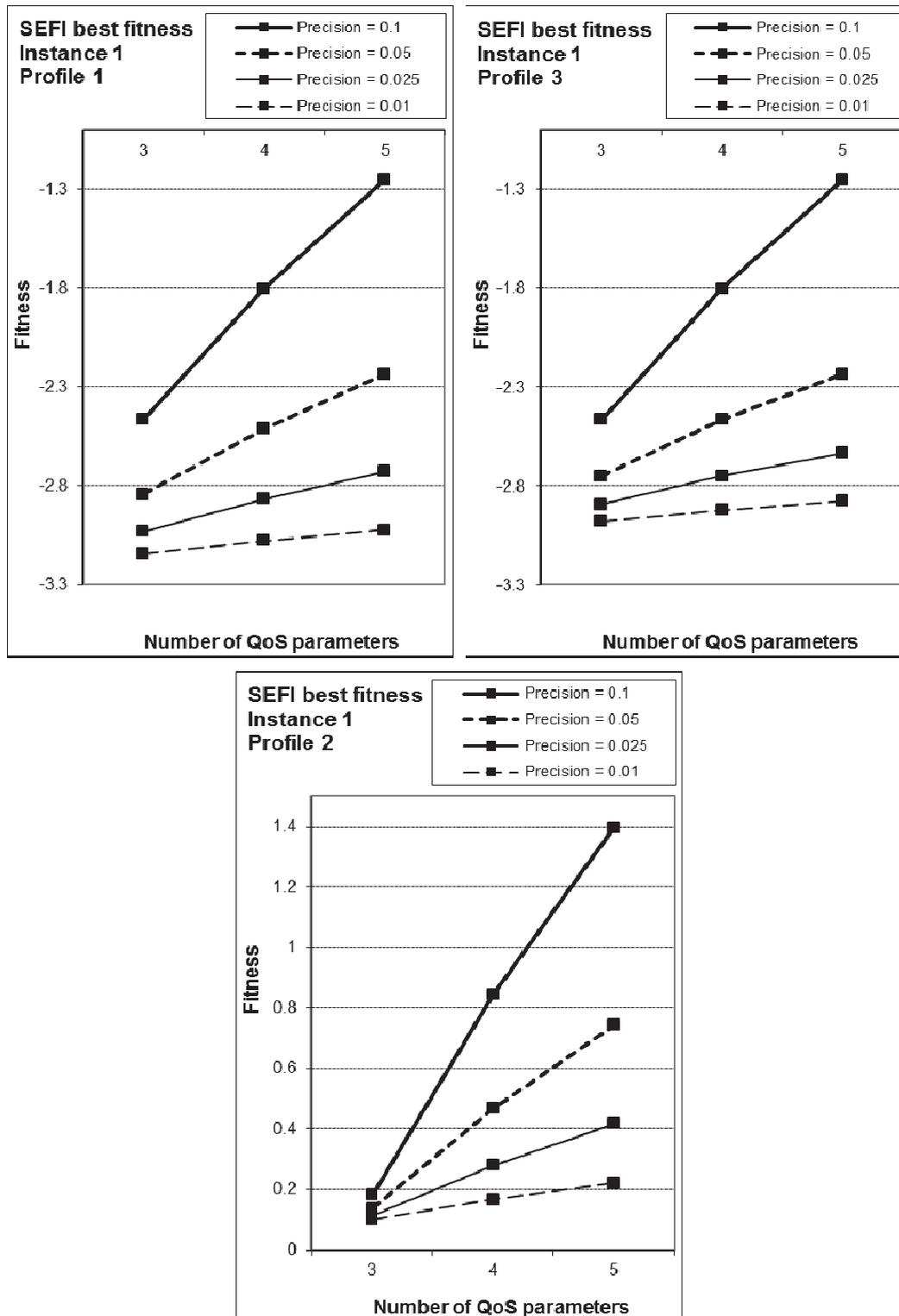


Figura 50. Mejor *fitness* encontrado por SEFI respecto al número de parámetros de calidad de servicio, para cada precisión. Datos obtenidos con la instancia #1, considerando los tres perfiles de usuario.

Las dos observaciones anteriores indican que **al aumentar el número de parámetros QoS y/o al aumentar la precisión de la búsqueda, se puede encontrar una red óptima distinta**. Esto demuestra la **necesidad de contar con heurísticas para la selección de una red óptima**, ya que cambiando la precisión o contando con más parámetros, la solución óptima puede cambiar. Esto avala el desarrollo del algoritmo SEFI, si bien también empuja a desarrollar nuevas heurísticas motivadas por factores que se explican más adelante.

Para las experimentaciones han de tenerse en cuenta los tiempos de computación, ya que se está en entornos de aplicaciones próximos al tiempo real. Las gráficas muestran el tiempo de computación para que se piense que aquellas condiciones de SEFI que impliquen tiempos superiores a 1 segundo deberían ser descartadas. En este sentido, se puede observar que ha de buscarse un equilibrio entre el número de parámetros QoS y el nivel de precisión  $h$ , ya que ambos factores inciden en el aumento del tiempo de computación.

No obstante, como la finalidad es seleccionar una red óptima, el análisis de las gráficas indica que **es más importante considerar un mayor número de parámetros QoS que aumentar el nivel de precisión**, ya que se encuentra una mejor red en todos los casos cuando se aumenta a 6 el número de parámetros QoS, mientras que solo en un caso se produce este hallazgo al aumentar el nivel de precisión (instancia #1, perfil #2, precisión 0,05). Incluso en este último caso, se puede observar que el aumento de la precisión no es excesivo (de  $h=0,1$  a  $h=0,05$ ), es decir, **se pueden considerar valores bajos de precisión sin menoscabar el hallazgo de la red óptima**.

En la Figura 52 se muestran las mismas gráficas pero para la instancia experimental #2. Para este conjunto de datos se ha considerado una precisión adicional mayor,  $h=0,005$ , ya que al tener menor número de parámetros QoS, los tiempos de computación son menores que para la instancia #1. Se puede observar que, en este caso, aumentar la precisión o aumentar el número de parámetros de calidad de servicio no implica encontrar una red mejor, por lo que la heurística SEFI, u otra similar, no sería de gran utilidad.

No obstante, es posible que, estableciendo ciertos perfiles de usuario más restrictivos (los perfiles #2 y #3 son poco restrictivos, ya que limitan solo 1 o 2 parámetros de calidad, respectivamente) que limiten la búsqueda de valores de más parámetros QoS, permita que aumentar la precisión en la búsqueda o el número de parámetros QoS pueda encontrarse una red óptima distinta.

Los resultados para ambas instancias experimentales no hacen sino resaltar una evidencia: en entornos dinámicos de conectividad inalámbrica heterogénea, las condiciones de las redes cambian constantemente, a veces de forma rápida, por lo que las heurísticas de selección deben estar preparadas para operar en cualquier escenario, donde la selección de la red óptima puede ser muy sensible al número de parámetros QoS y a la precisión de búsqueda (instancia #1), o no serlo tanto (instancia #2), o serlo a un cambio del perfil de usuario.

SEFI, por tanto, debe contemplar todas las posibilidades, y estar preparado para hacerles frente en tiempos de computación razonables.

### 6.6.3.5 Análisis estadístico de los resultados

Para la obtención de los resultados que se han presentado solo es necesaria una ejecución por cada experimento, donde se entiende por experimento la configuración de SEFI para unas condiciones de entrada dadas. No ha sido necesario repetir experimentos por medio de varias ejecuciones, por tanto no ha sido necesario realizar ningún tratamiento o análisis estadístico de los resultados.

La razón de no considerar tratamientos estadísticos radica en que no hay aleatoriedad en los distintos componentes o etapas de SEFI. En este sentido, no se ha programado ninguna función que pueda obtener resultados distintos para las mismas condiciones de entrada.

### 6.6.4 Conclusiones

A modo de resumen, se pueden listar las siguientes conclusiones:

- **A mayor precisión en la búsqueda y/o mayor número de parámetros QoS, mayor tiempo de computación.** Esto se debe a que el tiempo de computación es directamente proporcional al número de posibles soluciones generadas, número que aumenta cuando se incrementa el valor de  $NQ$  o se disminuye el de  $h$ . En algunos casos el incremento del tiempo de computación puede ser exponencial, haciendo inviable encontrar una solución en tiempos pequeños para cualquier valor de  $h$  o número de parámetros QoS. **Para asegurar la obtención de soluciones óptimas en tiempos pequeños, se consideran los límites impuestos en la Tabla 16** para SEFI, de forma que los tiempos de computación se mantengan por debajo de 1 segundo.
- **A mayor precisión, mejor *fitness* encontrado.** También, el número de combinaciones y el de **soluciones aumenta con la precisión y con el número de parámetros QoS.** El número de combinaciones es siempre mucho mayor que el número de soluciones. El número de soluciones tiende a disminuir cuanto mayor es el número de parámetros QoS y cuanto menor es la precisión de búsqueda.
- El tema de la precisión es importante, pero no contemplar más parámetros QoS puede llevar a conclusiones erróneas. Se ha demostrado que al aumentar el número de parámetros QoS y/o al aumentar la precisión de la búsqueda, se puede encontrar una red óptima distinta, pero que **es más importante considerar un mayor número de parámetros QoS que aumentar el nivel de precisión.** En este sentido, se pueden considerar valores algo más bajos de precisión sin menoscabar el hallazgo de la red óptima.

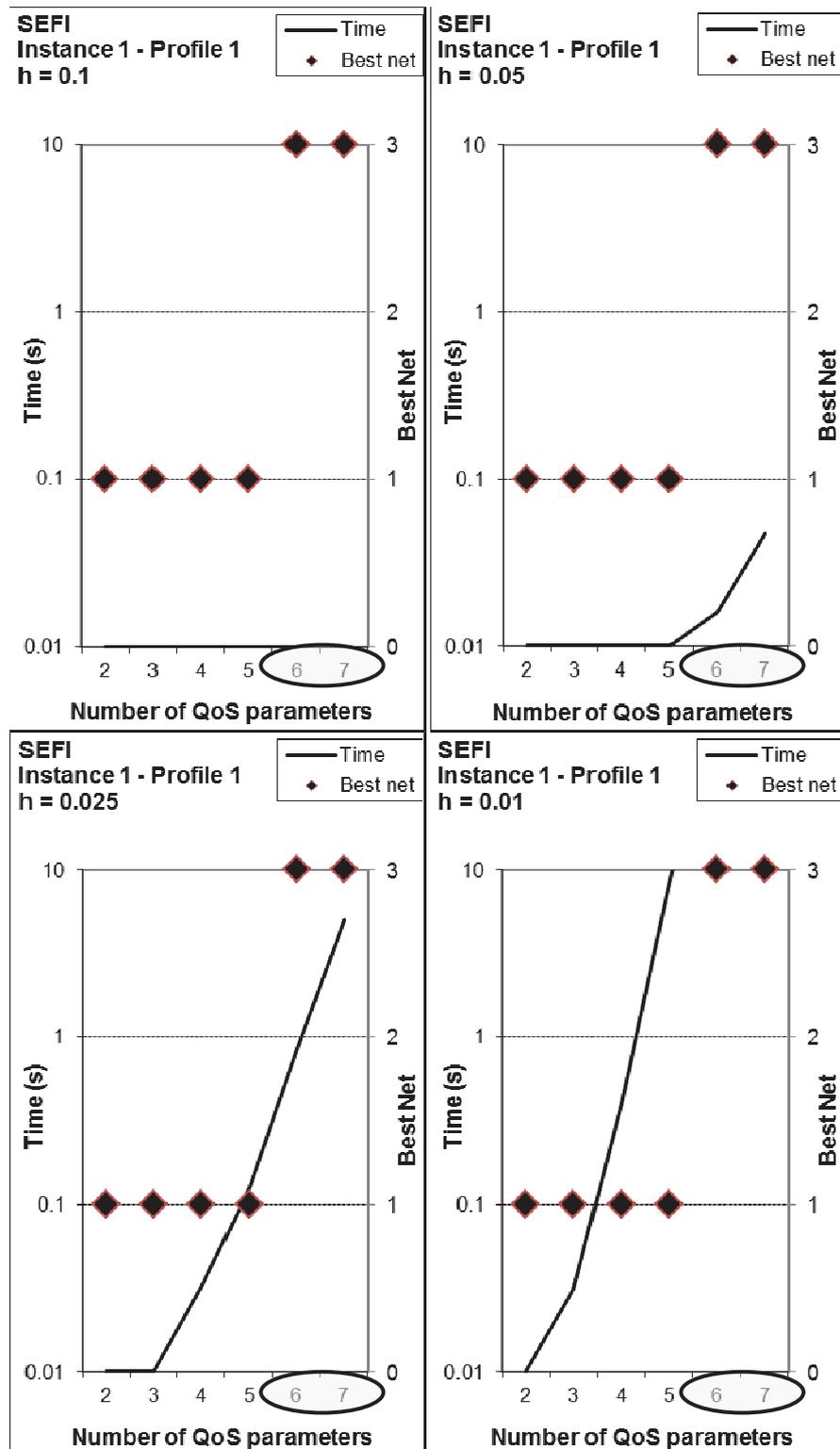


Figura 51. Tiempo de computación y mejor red encontrada por SEFI, para distinto número de parámetros QoS y precisiones distintas. Cada página de gráficas corresponde con los un perfil de usuario para la instancia #1, mientras que cada gráfica en la página corresponde con una misma precisión. Los círculos señalan datos significativos explicados en el texto.

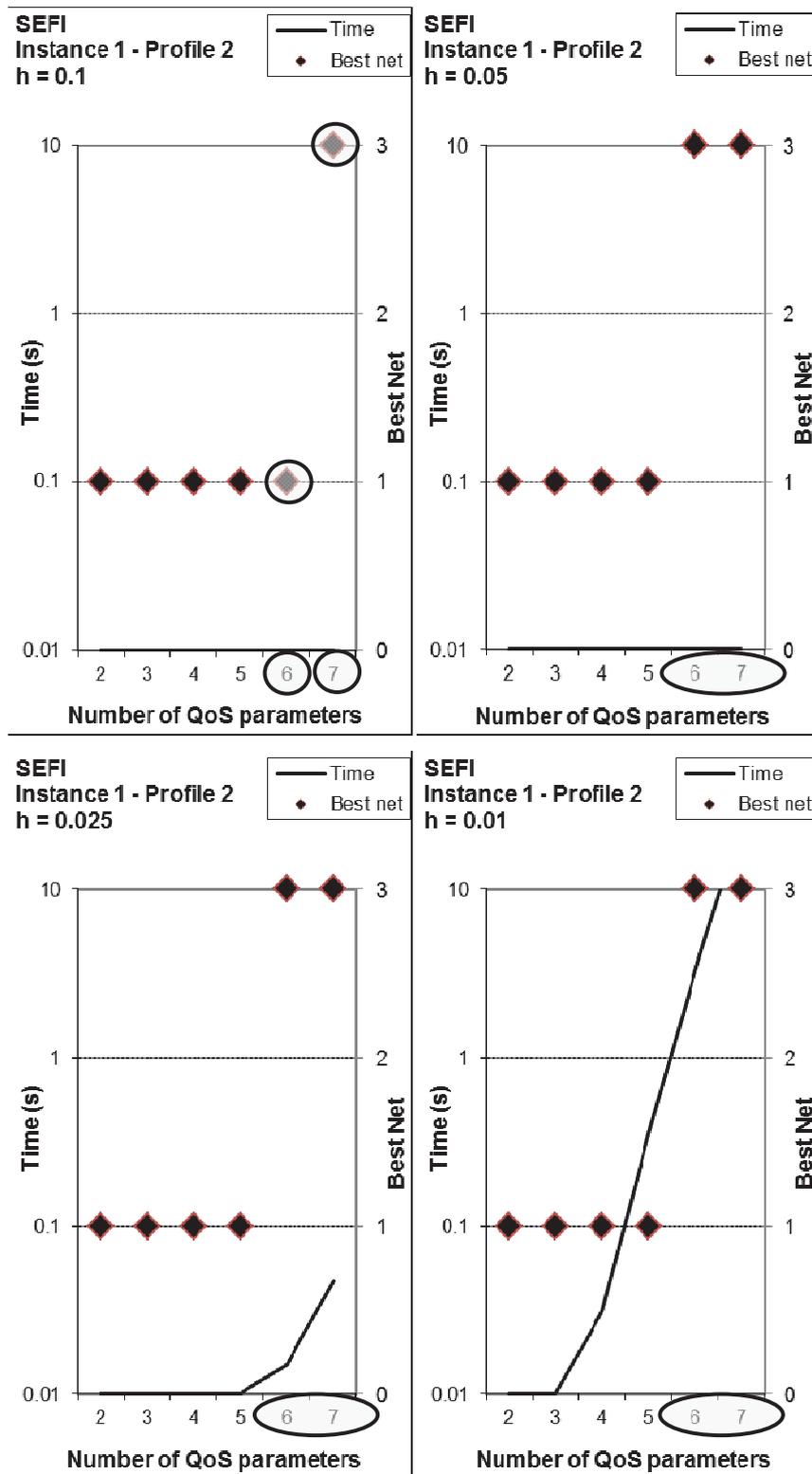


Figura 51. (continuación).

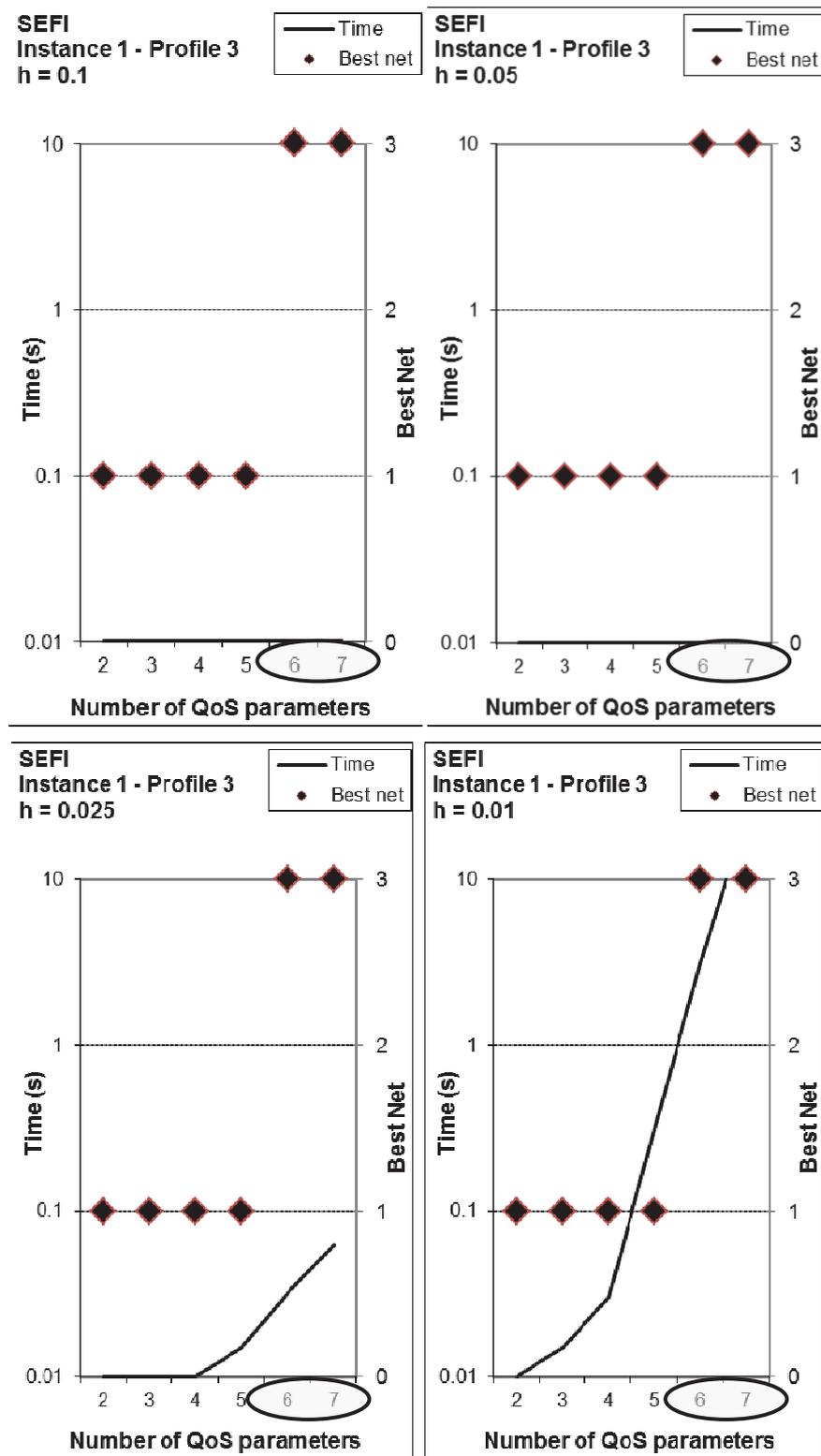


Figura 51. (continuación).

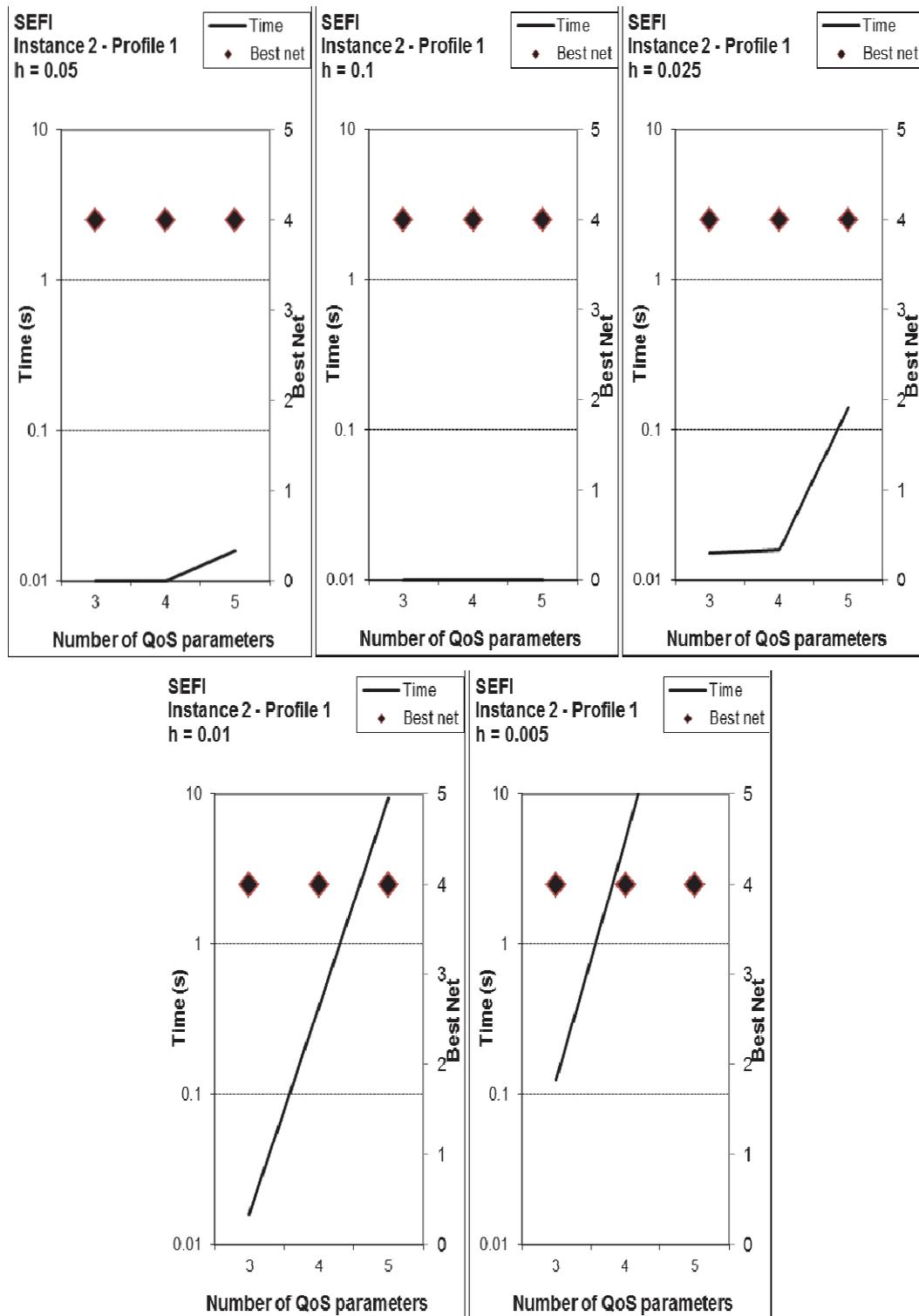


Figura 52. Tiempo de computación y mejor red encontrada por SEFI, para distintos números de parámetros de calidad de servicio considerados, y teniendo en cuenta precisiones distintas. Las gráficas en cada página corresponden a un perfil distinto para la misma instancia experimental #2, mientras que cada gráfica en la página ha sido generada con una precisión distinta.

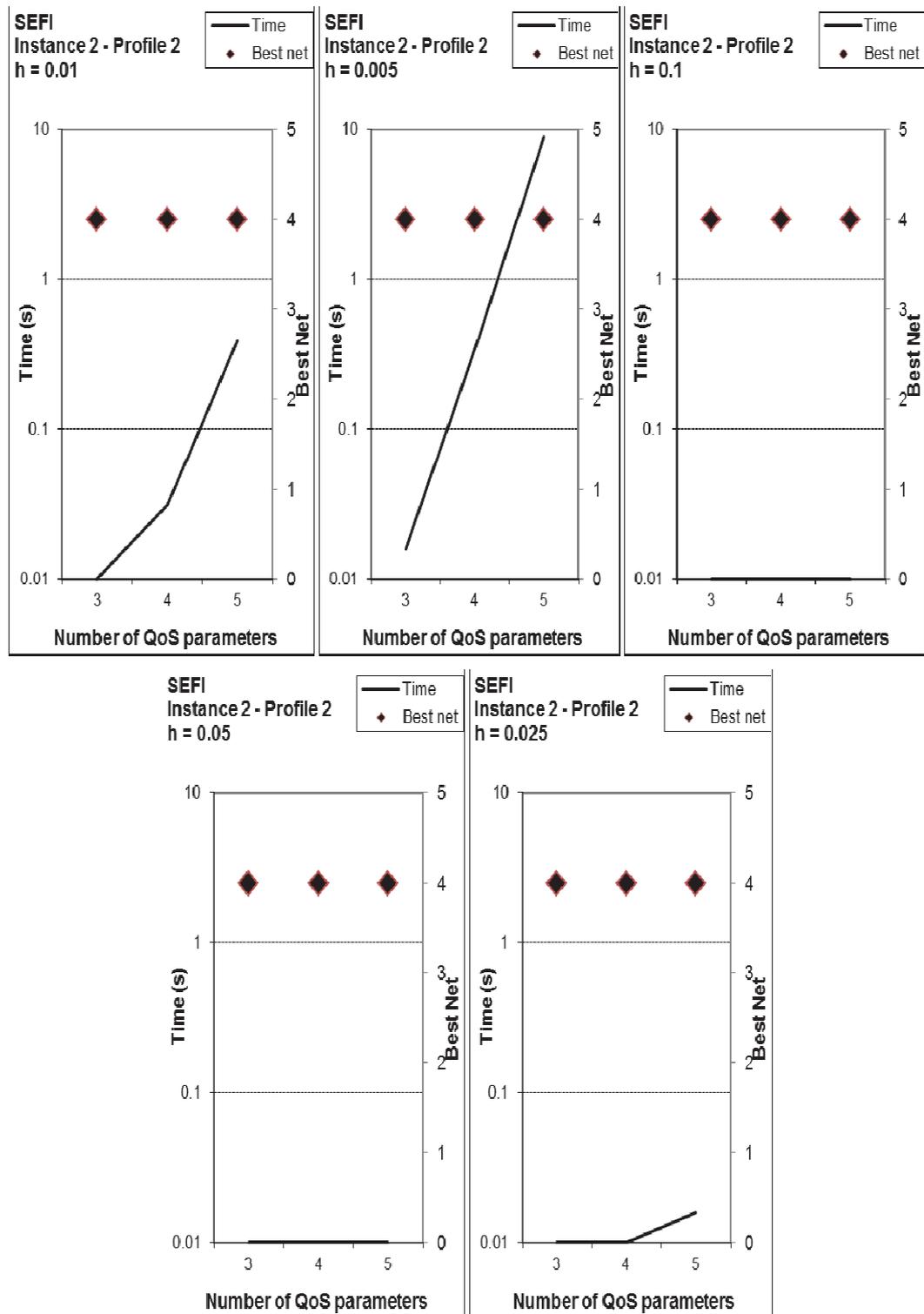


Figura 52. (continuación).

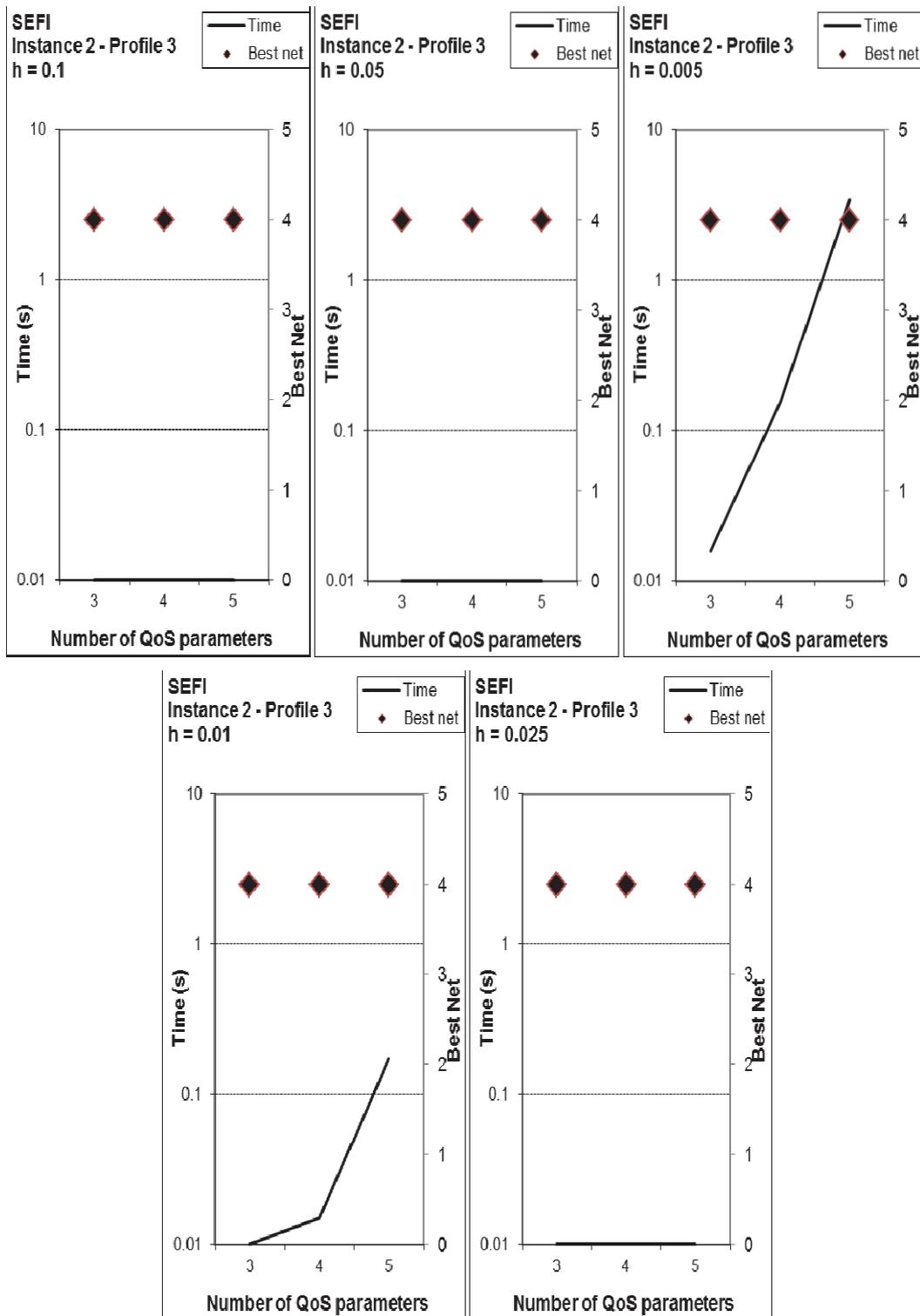


Figura 52. (continuación).

### 6.6.5 Claves para la concepción de una nueva heurística

Los anteriores resultados experimentales parecen indicar que la clave para desarrollar heurísticas de selección óptima de redes en procesos de VH está en aumentar el número de parámetros QoS.

Si bien esto es cierto, hay que tener en cuenta que, tal como puede observarse en la Figura 51, el aumento de este número de parámetros QoS incide en obtener tiempos de computación que pueden superar a lo que se considera tiempo real, por lo que para estos casos hay que considerar precisiones bajas. En apariencia, esto es una contradicción, ya que las precisiones bajas no favorecen el hallazgo de soluciones óptimas. Eso empuja, por tanto, a **idear heurísticas que, moviéndose en entornos bajos de precisión** (para poder considerar más parámetros QoS), **puedan encontrar soluciones óptimas en tiempo real**. Esta es la base para el desarrollo de las heurísticas evolutivas SEFISA y GAVH de esta tesis, que se explican más adelante.

Cualquier nueva heurística que se pueda plantear debe incorporar algún tipo de técnica mínimamente “inteligente”, pues el problema de optimización, tal como se ha planteado, encuentra las complejidades y restricciones que se han mostrado en los anteriores apartados. En este sentido, se destaca la siguiente afirmación, que motiva el esfuerzo en los trabajos que se han desarrollado para la implementación de las heurísticas: *“As new metrics and cost functions are developed, the implementation of a handoff policy becomes more important. Increased network and user interaction will increase the handoff latency. Thus, intelligent techniques must be developed to evaluate the effectiveness of the new algorithms, balanced against user satisfaction and network efficiency”* [McNair 2004].

## 6.7 Implementación hardware de SEFI

### 6.7.1 **Motivaciones**

Se propone un diseño digital basado en procesadores empotrados que responde a una de las actuales tendencias en computación móvil: **bajo coste, bajo consumo y buen rendimiento**. La Red de Excelencia en Compilación y Arquitecturas Empotradas de Alto Rendimiento, HiPEAC (The HiPEAC Network of Excellence on High Performance and Embedded Architecture and Compilation) especifica en qué consiste esta tendencia: *“People no longer only want more features and better performance, but are increasingly interested in devices with the same performance level at a lower price. [...] The limited processor performance also reduces power consumption and therefore improves mobility [...]. This trend also has an impact on software, as it now needs to be optimized to run smoothly on devices with less hardware resources [...] This trend is also leading to computers specifically designed to have extreme low power consumption”* [Duranton 2012].

El bajo coste, bajo consumo y buen rendimiento (una rápida obtención de soluciones óptimas) son requisitos deseables en un pequeño dispositivo electrónico móvil (tableta, teléfono, etc.) que se desplaza a lo largo de redes inalámbricas heterogéneas, como es el caso que se estudia en esta tesis. Por tanto, estos requisitos han motivado el diseñar de un microprocesador empotrado basado en hardware reconfigurable capaz de albergar y ejecutar el algoritmo SEFI.

Los escenarios dinámicos implican la necesidad de un procesamiento rápido. En este sentido, los procesadores empotrados basados en hardware reconfigurable (FPGAs) han demostrado ser una buena solución [Bodnar 2009][Zhou 2006]. Por otro lado, el requisito de bajo consumo eléctrico es muy importante, dada la alta demanda solicitada a la batería del terminal móvil (de limitadas capacidades) por parte de los distintos servicios que son utilizados por el usuario. Finalmente, cualquier solución hardware debe considerar el bajo coste económico de los componentes a causa de la competitividad de los distintos fabricantes.

### 6.7.2 **Tecnología y desarrollos**

Se ha considerado la tecnología de hardware reconfigurable basada en **FPGAs** (*Field Programmable Gate Array*) [Hauck 2008] por las ventajas que aporta: facilidad de programación mediante lenguajes de descripción hardware, alto grado de configurabilidad de microprocesadores empotrados, bajo consumo, reducido tamaño, bajo coste, coprocesamiento especializado, etc.

Todas estas características han propiciado el desarrollo de aplicaciones que se engloban en lo que se denomina Computación Reconfigurable de Alto Rendimiento (HPRC) [Buell 2007]. Las FPGA están desplazando la tecnología que antes (y aún ahora) se utilizaba ampliamente para el desarrollo de los sistemas de computación integrados en pequeños dispositivos: **ASICs** (*Application-Specific Integrated Circuits*). Por otro lado, la facilidad de programación y la rapidez del ciclo de diseño de los sistemas basados en FPGA hoy en día, propician

la preferencia de esta tecnología frente al ASIC. De esta forma, la tecnología hardware basada en FPGAs es una buena alternativa para muchas aplicaciones en el ámbito de las redes y comunicaciones.

Cabe destacar, dentro del área de la computación reconfigurable, las aplicaciones basadas en procesadores empotrados. Estos procesadores han sido desarrollados con el propósito de facilitar la construcción de soluciones fáciles de programar [Wolf 2007]. Las ventajas de utilizar un procesador empotrado para albergar y ejecutar el algoritmo SEFI residen en las ya apuntadas anteriormente: rapidez y facilidad de desarrollo algorítmico (códigos basados en C), buen rendimiento (tiempos de computación aceptables), bajo consumo, bajo coste, reducido tamaño (alto grado de integración en los dispositivos) y la reconfigurabilidad necesaria en algunos casos, como puede ser, por ejemplo, la actualización de *firmwares*.

En la Figura 53 puede observarse un sencillo esquema que indica la forma en que SEFI opera en el procesador empotrado, y cómo éste coexiste con los sistemas del dispositivo móvil que suministran al procesador la información necesaria para SEFI: redes encontradas, valores de los parámetros de calidad de servicio, etc.

Se ha utilizado el entorno ISE v13.3 y los dispositivos FPGA proporcionados por Xilinx [Xilinx 2011]. El diseño básicamente consiste en un microprocesador empotrado Microblaze con unidad de procesamiento en coma flotante, bus local y sistema operativo *standalone*. Microblaze es un procesador “blando”, esto es, completamente configurable por el usuario e implementable en FPGA. Las principales características de las implementaciones que se han hecho están listadas en la Tabla 17: S3 (basada en un dispositivo de bajo coste y consumo Spartan3E) y V5 (basada en un dispositivo de alto rendimiento Virtex5).

Ambas implementaciones utilizan tarjetas de prototipado comerciales (Figura 54), en las que los dispositivos FPGA usados no necesitan ni ventiladores ni disipadores de calor (que ocupan mucho espacio), y los periféricos o dispositivos externos a la FPGA no son necesarios para nuestros propósitos, posibilitando circuiterías muy reducidas, atractivas para terminales móviles.

La implementación S3 está orientada a un desarrollo de pequeño tamaño, de bajo consumo y muy económico. Utiliza un reloj a 50 MHz y ofrece un consumo de 0,01 W (estimado a partir de la herramienta XPower Analyzer), pero requiere de una memoria externa porque los recursos internos de la FPGA no son suficientes para albergar también el código de SEFI. No obstante, el chip de memoria externa RAM puede incluirse fácilmente en cualquier terminal móvil dado su reducido tamaño. El área ocupada indica que el microprocesador puede implementarse en este dispositivo tan económico (20 dólares para el dispositivo xc3s250e).

La implementación V5 representa una solución de alto rendimiento, que utiliza un ciclo de reloj más corto (mayor velocidad del circuito) y donde los recursos internos de la FPGA son suficientes para albergar al microprocesador junto con el código de SEFI. Esta implementación de alto rendimiento puede ser interesante para algunos tipos de aplicaciones y desarrollos.

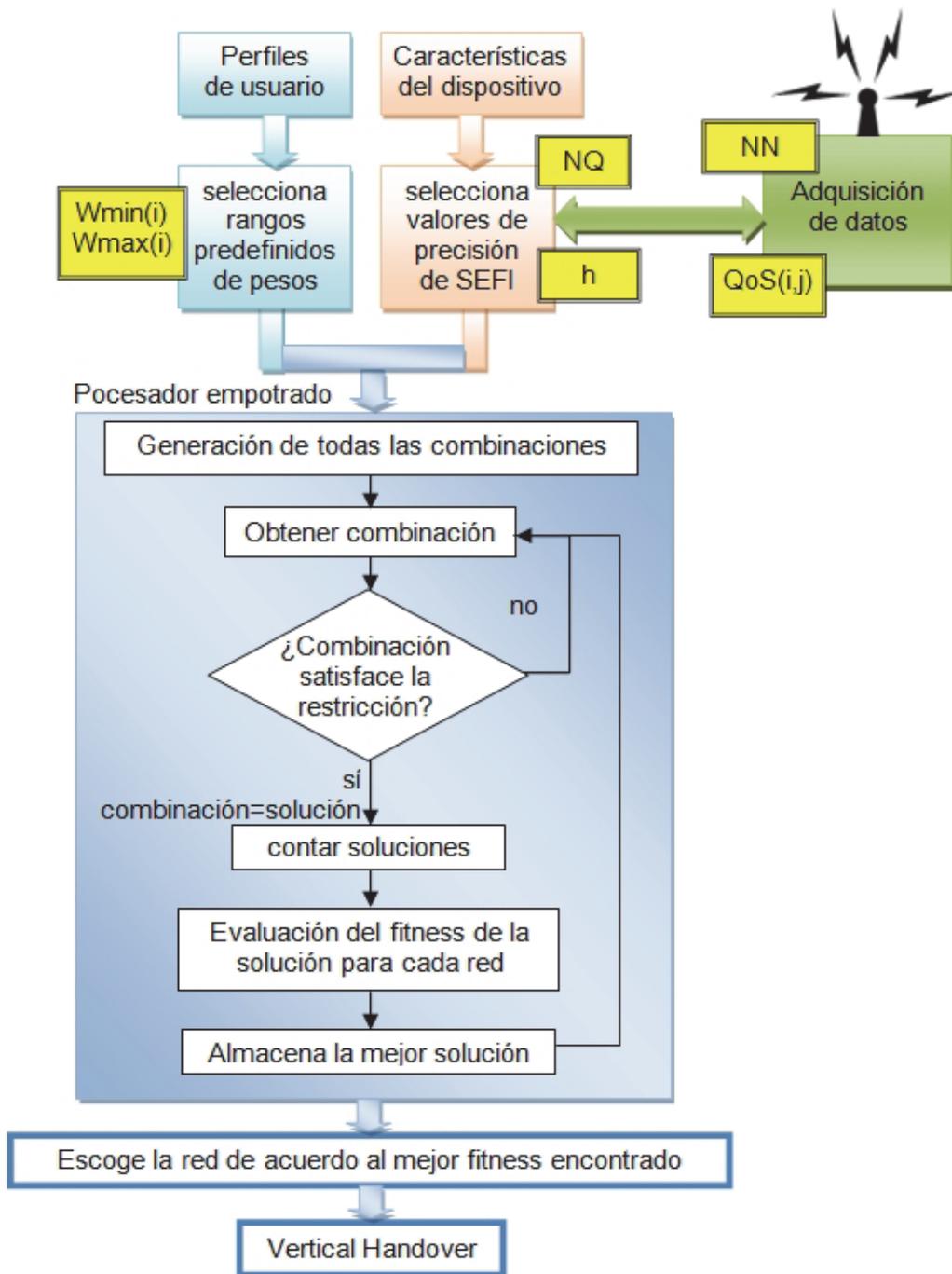


Figura 53. Integración y comportamiento de SEFI en la arquitectura hardware del dispositivo inalámbrico móvil.

	<b>S3</b>	<b>V5</b>
Tarjeta de prototipado	Digilent Nexys2-500	Xilinx XUPV505-LX110T
FPGA	Spartan 3E: xc3s500e-fg320-4	Virtex5: xc5vlx110t-ff1136-1
Procesador	Microblaze con FPU y bus PLB	Microblaze con FPU y bus PLB
Memoria	Externa	Interna / 256 KB
Sistema operativo	Standalone	Standalone
Frecuencia de reloj	50 MHz	125 MHz
Recursos FPGA ocupados	43%	10%
Consumo energético	0.097 W	1.34 W

Tabla 17. Características de las dos implementaciones hardware desarrolladas para el microprocesador empotrado que alberga y ejecuta SEFI.

### 6.7.3 Resultados

Los resultados arrojados por las implementaciones (y los análisis y conclusiones derivadas), son los mismos que los ya apuntados anteriormente al analizar el algoritmo SEFI. Los resultados que ahora se comentan tienen que ver con los tiempos de computación, ya que las medidas de tiempos de SEFI antes analizadas fueron hechas sobre ejecuciones en microprocesadores comerciales de bajo rendimiento, como es Intel Core2 E6750 2,6GHz.

Los microprocesadores empotrados MicroBlaze ofrecen ahora un bastante menor rendimiento (50 MHz y 125 MHz respecto a 2,6 GHz), lo cual se traduce en tiempos de computación muy superiores. Esto hace necesario ajustar el funcionamiento de SEFI según la Tabla 16 a los nuevos rendimientos. De esta forma, los tiempos medidos en las implementaciones hardware son mostrados en la Figura 55. Analizando estos datos, se pueden establecer los límites de funcionamiento de SEFI según los valores mostrados en la Tabla 18.

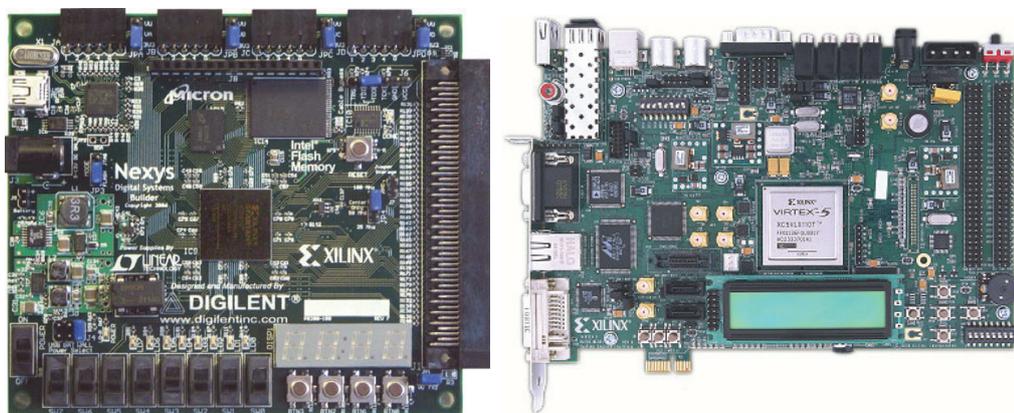


Figura 54. Tarjeta de prototipado Digilent Nexys (izquierda) con FPGA Spartan xc3s500e y Digilent XUPV5 (derecha), con FPGA Virtex5 xc5vlx110t. La primera es una solución económica, de reducido tamaño y bajo consumo, mientras que es una solución de alto rendimiento.

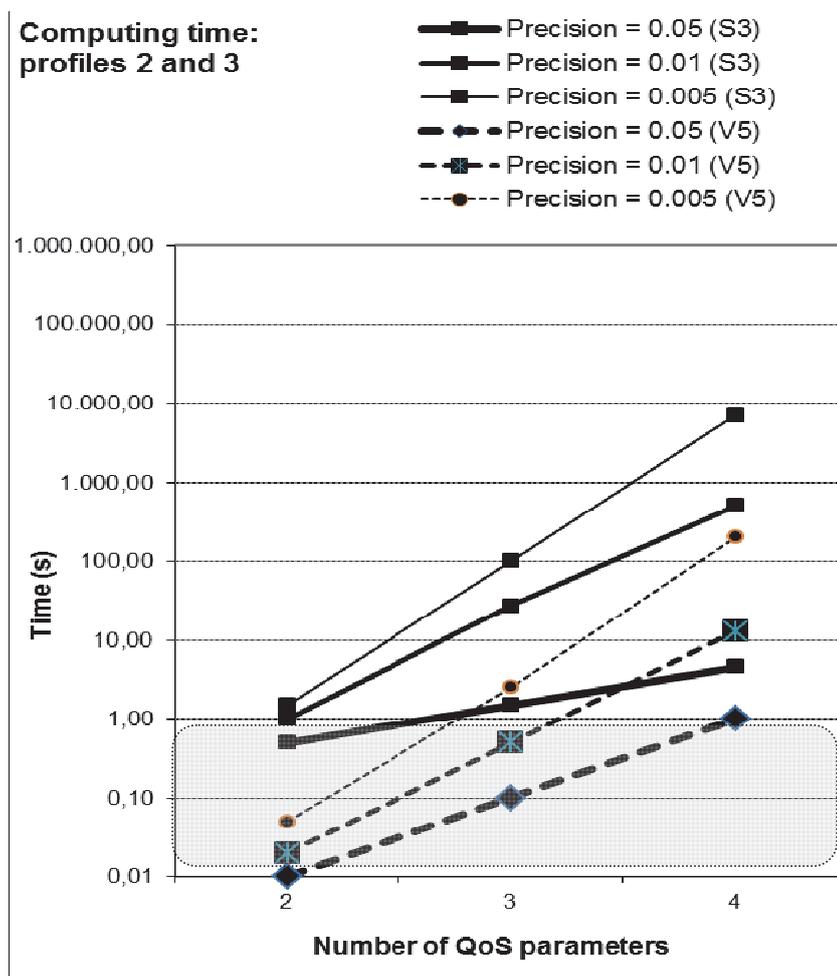


Figura 55. Tiempos de computación de SEFI en las implementaciones hardware S3 y V5 para aplicaciones de SEFI que cuentan con perfiles de usuario. La zona sombreada indica lo que se considera como tiempos de respuesta aceptable en entornos de movilidad inalámbrica (1 segundo).

**Implementación S3**

NQ	h
2	$h \geq 0.005$
3	$h \geq 0.05$

**Implementación V5**

NQ	h
2	$h \geq 0.001$
3	$h \geq 0.005$
4	$h \geq 0.01$

Tabla 18. Valores de precisión ajustados para las implementaciones S3 y V5.

#### **6.7.4 Conclusiones**

Los microprocesadores empotrados en FPGAs basados en la arquitectura Microblaze ofrecen bajo rendimiento, pero aceptable para algunos rangos de precisión de SEFI y cuando se consideran pocos parámetros de calidad de servicio con rangos predefinidos de usuario (que reducen los tiempos de computación, y que son útiles para aplicaciones específicas).

En este trabajo de tesis se ha podido demostrar que el algoritmo SEFI es fácilmente implementable en esta tecnología (gracias a su código en lenguaje C) y ofrece ventajas como las siguientes:

- Consumo energético muy reducido (con el consiguiente aumento del tiempo de vida de las baterías).
- Alto grado de integración (para dispositivos de reducido tamaño, el chip Spartan3E es muy adecuado).
- Bajo coste económico.

El bajo rendimiento de computación que ofrecen los dispositivos con este tipo de microprocesadores basados en hardware reconfigurable no impide considerar la implementación de SEFI, pues el rendimiento es bastante aceptable en terminales móviles para aplicaciones donde el número de parámetros de calidad de servicio es reducido. Además, fijar los límites de los valores para las posibles combinaciones ayuda a no demandar más recursos computacionales.

Es decir, ciertas aplicaciones concretas del terminal móvil no necesitan de muchos parámetros de calidad. Por ejemplo, para aplicaciones de transmisión multimedia solo es interesante el ancho de banda (para evitar retardos en la entrega de la señal que ocupa mucho tamaño), mientras que para aplicaciones de comunicación verbal solo son interesantes dos variables, principalmente: el retardo de la señal y el coste económico del servicio.

## 6.8 Algoritmo SEFISA

En este apartado se explica la heurística que se ha desarrollado a partir de las motivaciones expuestas en la sección 6.6.5.

### 6.8.1 Características

SEFISA es el término por el cual se nombra a la heurística adaptativa que se propone en esta tesis, y que está basada en el algoritmo de Enfriamiento Simulado, *Simulated Annealing (SA)*, con el fin de otorgar a SEFI un carácter adaptativo que permita incrementar su eficiencia.

El algoritmo SA [Kirkpatrick 1983][Cerny 1985] se inspira en el proceso de enfriamiento de un metal en el que se busca una estructura final de mínima energía, y cuyo proceso está formado de etapas sucesivas que determinan estructuras cada vez de menor energía. En su formulación general, SA busca un estado de mínima energía dentro de un espacio de búsqueda de soluciones definido; de esta forma, una vez encontrada una solución óptima, se vuelve a buscar otra en un nuevo espacio de búsqueda, esta vez más reducido, y centrado en la solución óptima encontrada anteriormente. La reducción sucesiva de cada espacio de búsqueda está determinada por un factor de reducción.

La versatilidad de este algoritmo permite su adaptación a muchos problemas de optimización, con variantes en cuanto al método de búsqueda, reducción del espacio de búsqueda, criterios de parada, etc. En el caso contemplado en esta tesis, SA es una fuente de inspiración para la heurística SEFISA, especialmente en su vertiente adaptativa, que sigue la filosofía de la reducción de los espacios de búsqueda centrados en las sucesivas soluciones óptimas encontradas. En la Figura 56 se puede observar el pseudo-código que resume el funcionamiento básico de SEFISA, el cual es explicado más detalladamente en la siguiente sección.

```
1: Inicializacion: precisión de búsqueda (h) y perfil de usuario
2: Establecer espacios de búsqueda para cada peso (Ei)
3: Establecer factor de reducción (Red)
4: IdGeneracion = 0
5: DO
6:     Ejecutar SEFI
7:     Localizar solución óptima
8:     Ei = Ei / Red
9:     Calcular los nuevos espacios de búsqueda:
10:         centrar en pesos correspondientes a la solución óptima
11:         ajustar los límites locales según las restricciones
12:     IdGeneracion++
13: WHILE NOT criterio de parada (forzoso o predeterminado)
```

Figura 56. Pseudo-código resumido de SEFISA.

## 6.8.2 Implementación

SEFISA ha sido programado para seguir el siguiente modo de funcionamiento, tal como se puede visualizar en la Figura 57:

- Inicialmente, la primera generación (generación #0) ejecuta el algoritmo SEFI de un modo ordinario, donde los espacios de búsqueda para los pesos de cada parámetro QoS  $w_i$ , están determinados por los límites  $u_{min,i}$  y  $u_{max,i}$ , los cuales determinan sus tamaños  $D_i$ , y que son impuestos por las preferencias del usuario o el perfil de la aplicación.

En esta primera generación se establece el valor constante  $div$ , el cual establece la división del espacio de búsqueda más pequeño, es decir, determina los  $div$  valores de este espacio de búsqueda que serán utilizados para generar las posibles combinaciones.

El valor  $div$  se define al comienzo de SEFISA y es invariable durante todas las generaciones sucesivas. Esta división del espacio de búsqueda propicia poder calcular el correspondiente valor de precisión de búsqueda,  $h$ .

El cálculo de  $h$  se hace de la siguiente forma: se determina el valor  $D_i$  mínimo en la generación,  $D_{min}$  (en el ejemplo de la Figura 57 corresponde al peso  $w_1$ ) y se divide por  $div$ , resultando un valor de  $h$  que se fija para todos los espacios de búsqueda de todos los pesos  $w_i$ , de forma que algunos espacios tendrán el mismo o mayor número de posibles valores para generar combinaciones que el que tiene el espacio correspondiente a  $D_{min}$ .

De esta forma,  $h$  es fijo para una misma ejecución de SEFI (para una misma generación, para todos sus espacios de búsqueda) pero distinto en cada generación, dependiendo del valor  $D_{min}$  y  $div$ .

- Una vez encontrado el *fitness* óptimo en la primera ejecución de SEFI, se localiza la combinación de pesos que le corresponde (solución óptima). Cada uno de estos pesos centrará un nuevo espacio de búsqueda para la siguiente generación de SEFISA, es decir, para la siguiente ejecución de SEFI, espacio que además quedará reducido en su extensión.

La reducción del espacio de búsqueda actual  $D_i$  está determinada por la constante  $red$ , que recibe el nombre de factor de reducción, tal como se ve en la ecuación (35), y que usualmente se tomará como 2 (los sucesivos espacios de búsqueda se reducen a la mitad). Este nuevo tamaño del espacio de búsqueda,  $D'_i$  hará que se calculen los nuevos límites de búsqueda  $v_{min,i}$  y  $v_{max,i}$ , teniendo en cuenta que, para la primera generación de SEFISA, el espacio de búsqueda estaba determinado por  $u_{min,i}$  y  $u_{max,i}$ .

$$(35) \quad D'_i = \frac{D_i}{red} : ((D_i = v_{max_i} - v_{min_i}) \wedge (D_{i,o} = u_{max_i} - u_{min_i}))$$

donde:

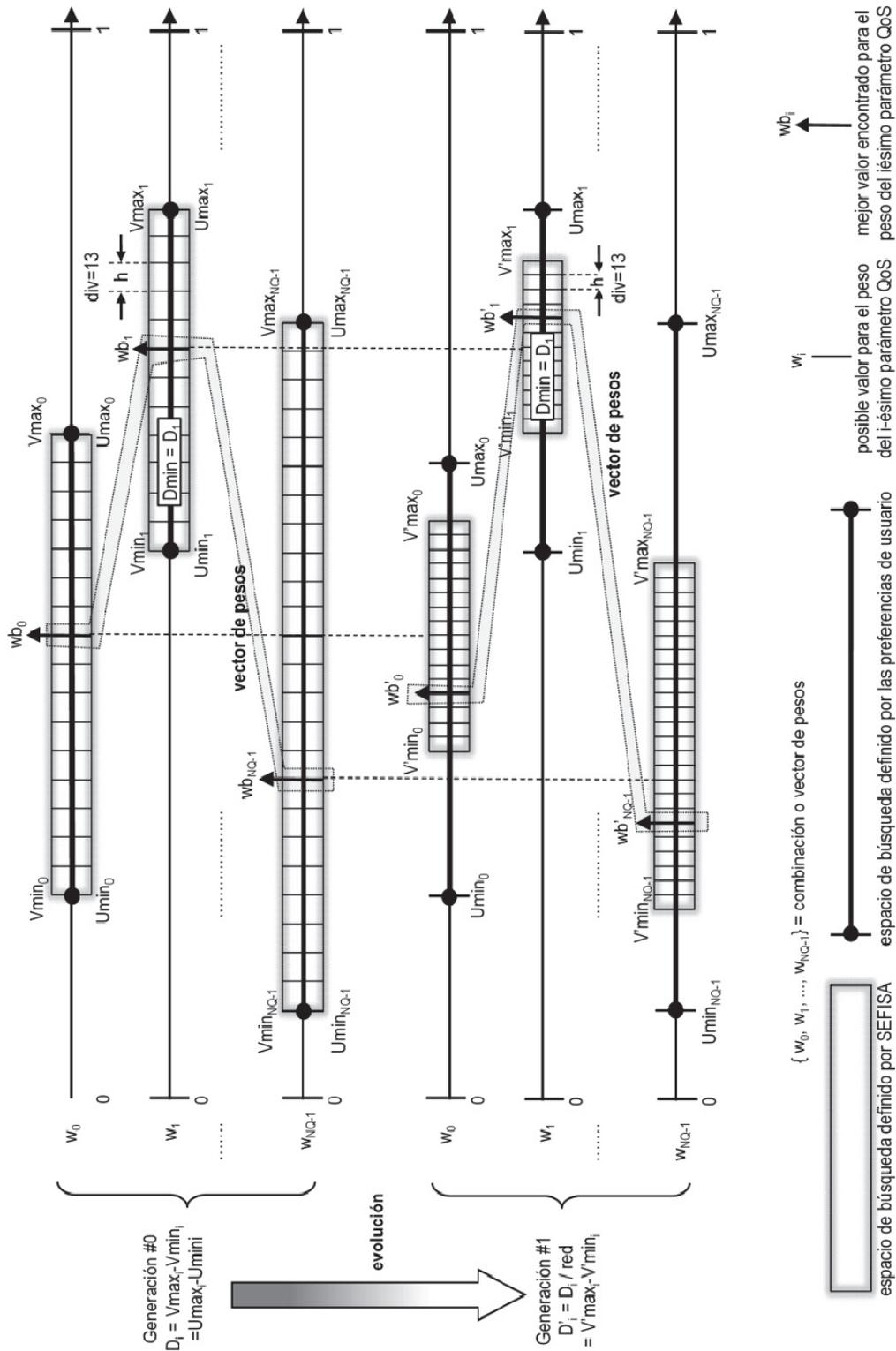


Figura 57. SEFISA ejecuta SEFI en generaciones sucesivas, donde cada generación ve reducido el espacio de búsqueda de soluciones, dependiendo del factor de reducción, de los límites impuestos por las preferencias de usuario (perfiles) y de los valores óptimos encontrados en la generación anterior, que son utilizados para centrar los nuevos espacios de búsqueda.

- $D_i$  = tamaño del espacio de búsqueda para el  $i$ -ésimo parámetro QoS.
- $D'_i$  = nuevo tamaño del espacio de búsqueda para el  $i$ -ésimo parámetro QoS (generación posterior a la actual).
- $red$  = factor de reducción.
- $\{u_{min,i}, u_{max,i}\}$ : límites, definidos según las preferencias del usuario o el tipo de aplicación, del espacio de búsqueda inicial (primera generación de SEFISA) para el peso del  $i$ -ésimo parámetro QoS.
- $\{v_{min,i}, v_{max,i}\}$ : límites del espacio de búsqueda para el peso del  $i$ -ésimo parámetro QoS en una determinada generación, calculados a partir del actual tamaño del espacio de búsqueda y del centro del mismo, que corresponde con el  $w_i$  óptimo encontrado en la generación anterior. Estos límites están incluidos, siempre, dentro del espacio de búsqueda inicial definido por  $\{u_{min,i}, u_{max,i}\}$ .

Una vez calculado el nuevo tamaño del espacio de búsqueda, se determinan los límites del mismo. Para este cálculo hay que tener en cuenta las posibles situaciones particulares que pueden producirse.

Un caso que puede presentarse es cuando el nuevo espacio de búsqueda excede los límites permitidos por el perfil de usuario. En la Figura 58 se muestra un ejemplo de este caso. Supóngase una generación #J donde el parámetro  $w_i$  tiene un espacio de búsqueda  $D^{(J)}_i$  definido por el rango  $\{Vmin^{(J)}_i, Vmax^{(J)}_i\}$ , el cual estará incluido siempre dentro del rango definido por  $\{Umin_i, Umax_i\}$  (este último rango es independiente de la generación, pues es establecido inicialmente y no cambia a lo largo de la evolución de SEFISA). Para la siguiente generación #J+1, el rango es reducido a la mitad,  $D^{(J+1)}_i = D^{(J)}_i/2$ , lo cual hace (en este ejemplo), que el límite  $Vmin^{(J+1)}_i$  quede por debajo del límite mínimo posible,  $Umin_i$ , por lo que hay que hacer un ajuste de este nuevo límite. Las condiciones de límites y casos de ajuste de muestran en las ecuaciones (36) a (39).

$$(36) \quad V^{(J+1)}_{min_i} = wb^{(J)}_i - D^{(J+1)}_i / 2$$

$$(37) \quad V^{(J+1)}_{max_i} = wb^{(J)}_i + D^{(J+1)}_i / 2$$

$$(38) \quad \text{Si } (V^{(J+1)}_{min_i} \leq Umin_i) \rightarrow V^{(J+1)}_{min_i} = Umin_i$$

$$(39) \quad \text{Si } (V^{(J+1)}_{max_i} \geq Umax_i) \rightarrow V^{(J+1)}_{max_i} = Umax_i$$

Hay otras casuísticas que también pueden producirse y que dan lugar a la necesidad de realizar otros pequeños ajustes, principalmente debido a las nuevas precisiones  $h$  para SEFI, que pueden no coincidir con las divisiones del espacio de búsqueda, o exceder la precisión de los tipos de datos definidos en C. Para estos y otros casos, SEFISA contempla las acciones de ajuste pertinentes que, por no extender en exceso esta memoria de tesis, no se detallan. No obstante, estas funciones de ajuste se pueden encontrar en el código fuente de SEFISA.

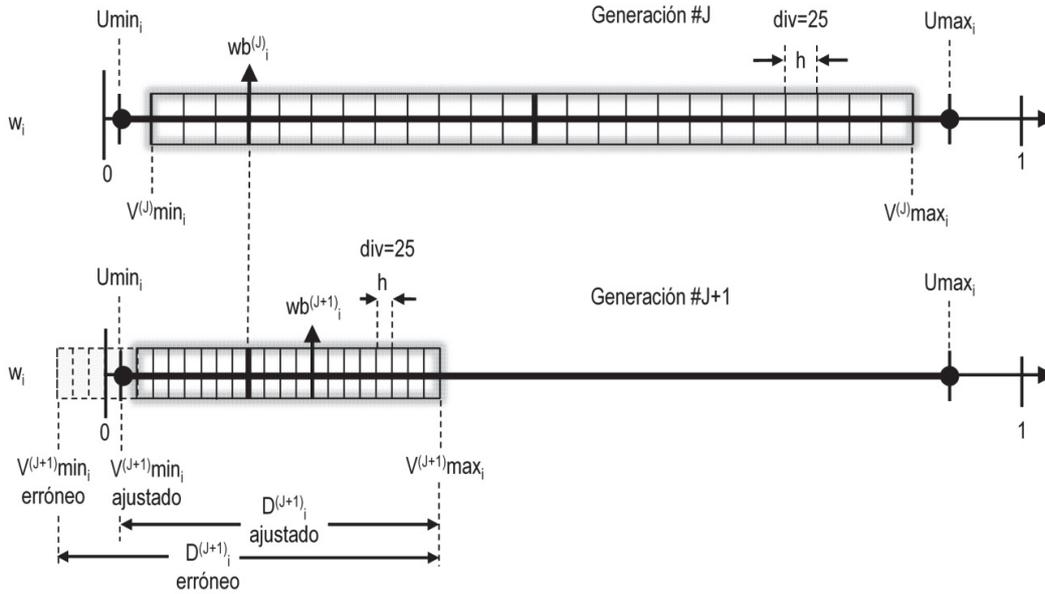


Figura 58. Caso en el que el espacio de búsqueda para  $w_i$  en la generación #J, al reducirse a la mitad en la generación #J+1, excede los límites establecidos por el perfil de usuario e, incluso, el mínimo posible, 0. En casos como éste han de re-ajustarse los límites que son erróneos; esto producirá que, si bien  $h$  sigue siendo el mismo, se reduzca el número de posibles valores para generar combinaciones con  $w_i$ .

Una vez determinados los límites del espacio de búsqueda para cada peso  $w_i$ , se aplica SEFI, generando todas las posibles combinaciones de pesos (conjunto  $NC$ ), y hallando todas las posibles soluciones (conjunto  $NS$ ) que satisfacen la restricción, la cual se especifica ahora con mayor exactitud en las ecuaciones (41)(40). Como ya se comentó anteriormente, el número de soluciones es siempre inferior al de combinaciones (41).

$$(40) \quad \sum_{i=0}^{i=NQ-1} w_i = 1 : \\ \left( (v_{min_i} \leq w_i \leq v_{max_i}) \wedge (0 \leq u_{min_i} \leq v_{min_i} \leq v_{max_i} \leq u_{max_i} \leq 1) \right)$$

$$(41) \quad NS \subseteq NC$$

donde

- $NQ$  = número de parámetros QoS.
- $w_i$  = peso para el  $i$ -ésimo parámetro QoS.
- $NC$  = número de combinaciones de pesos distintas para una precisión  $h$  determinada.
- $NS$  = número de combinaciones de pesos que satisfacen la restricción, esto es, número de posibles soluciones.

- Al aplicar SEFI ahora a la nueva generación, se obtiene un nuevo *fitness* mínimo junto con su correspondiente combinación de pesos, que son los pesos óptimos. De nuevo, los valores de estos pesos óptimos centrarán unos espacios de búsqueda aún más reducidos para la siguiente generación de SEFISA.

SEFISA puede tener, en teoría, muchas generaciones, pero según los experimentos efectuados y las restricciones del ámbito de aplicación de la heurística, no van más allá de 4 generaciones. En este sentido, cabe hablar de unos **criterios de parada**, que pueden ser **forzosos** o **predefinidos**:

- Criterios de parada **predefinidos**. Son criterios programados en el algoritmo, que definen situaciones previstas, que tarde o temprano se producirán, y que cuando se alcanzan abortan la evolución:

- Tiempo de computación.

Dado que el ámbito de aplicación de SEFISA es el de los terminales inalámbricos móviles, como ya se ha dicho anteriormente, ha de tenerse en cuenta una restricción en el tiempo de computación para que sea cercano al tiempo real, establecido en 1 segundo. Por tanto, cuando el tiempo de computación supere este umbral, SEFISA se parará, independientemente de la generación que haya alcanzado hasta el momento.

- Precisión alcanzada.

La parada del algoritmo se produce cuando la precisión de la búsqueda de soluciones puede originar resultados incorrectos. Hay varias causas que pueden provocar estos problemas de resolución. Las principales derivan del tipo de datos utilizados en el código C de SEFISA. El valor de precisión  $h$  (*float*) se digitaliza en la variable  $pu$  (*unsigned int*) para generar los valores de los pesos mediante ciertos mecanismos; después se utiliza la variable intermedia  $huf$  (*float*) para evitar pérdidas de precisión en esta digitalización, y posteriormente se asigna su valor a la variable  $hu$  (*unsigned int*) que es el valor digitalizado exacto para generar los valores de los pesos. Como la variable  $pu$  no puede exceder de la precisión de su tipo de datos (4.294.967.295), se aplica la restricción genérica de que  $0,5 < h < 0,0000000001$ , es decir, se fija un umbral de precisión en  $10^{-10}$ .

Pues bien, en las sucesivas generaciones de SEFISA, y como consecuencia de las reducciones de los espacios de búsqueda y sus divisiones para generar  $h$  más pequeños, entre otros factores, se pueden estar generando valores para  $h$  inferiores al umbral de precisión  $10^{-10}$ . En estos casos el algoritmo aplica una parada para finalizar en la generación actual.

- Criterios de parada **forzosos**. De momento se ha previsto un solo criterio de este tipo. Se trata de criterios de parada generados por situaciones no previstas, pero que de producirse, abortan la evolución del algoritmo:

- Ausencia de soluciones:

Si en una determinada generación no se encuentran soluciones, el algoritmo SEFISA aborta su evolución. En algunos casos la ausencia de soluciones se constata de forma muy rápida al ejecutarse el algoritmo, lo

cual puede permitir a SEFISA tener tiempo suficiente como para poder reiniciarse con otros parámetros de entrada (como la precisión, por ejemplo), que pueda ofrecer al algoritmo un mejor rendimiento.

### 6.8.3 Resultados experimentales

Se han realizado numerosas ejecuciones de SEFI teniendo en cuenta:

- Las dos instancias experimentales, con sus tres perfiles de usuario.
- Valor del factor de reducción *red* fijado en 2.
- Valores del factor de división *div*: 3, 5, 10, 20, 50, 100, 150.
- Número de parámetros de calidad de servicio: 3, 4 y 5.

#### 6.8.3.1 Tiempo de computación

A lo largo de los numerosos experimentos efectuados, se ha vuelto a constatar, como ya se hizo con SEFI, que el tiempo de computación (que es proporcional al número de combinaciones generadas y evaluadas) aumenta con el número de parámetros de calidad de servicio y con el grado de precisión de la división de los espacios de búsqueda. Eso produce que si el número de parámetros QoS es alto (5), no se pueda llegar a utilizar factores de división altos (como 50, 100 o 150), y si por el contrario dicho número es bajo, se pueda experimentar con todos los valores del factor de división.

#### 6.8.3.2 Mejora de la solución óptima a lo largo de las generaciones

En la Figura 59 se muestra una gráfica representativa de la evolución de la solución óptima encontrada a lo largo de las generaciones de SEFISA.

Esta evolución produce la obtención de valores de *fitness* cada vez más bajos (que podrían corresponder a distintas redes de acceso óptima). Esto demuestra el carácter evolutivo de la heurística que, basada en *Simulated Annealing*, fundamenta su evolución en:

- La **adaptación** de los espacios de búsqueda: éstos son reducidos sucesivamente y son centrados en los pesos óptimos que fueron encontrados en la generación anterior.
- La **búsqueda directa** de las soluciones (**técnica exacta**) dentro de los espacios de búsqueda. Esta búsqueda es efectuada mediante el algoritmo SEFI.

La gráfica de la Figura 59 ha sido obtenida a partir de la ejecución de SEFISA para la instancia #1 (4 redes de acceso inalámbricas) con 5 parámetros QoS y un valor *div*=3, siendo representativa de los muchos experimentos realizados, a partir de los cuales se sacan las siguientes conclusiones:

- En general, **SEFISA evoluciona encontrando cada vez mejores soluciones óptimas**, si bien se constata que con frecuencia se encuentran evoluciones que estancan la solución óptima en las últimas generaciones.

- La **obtención de óptimos sucesivos** (ausencia de estancamiento en la obtención de la solución óptima) se da con más frecuencia para **valores bajos del factor de división  $div$**  (3 y 5).
- El **criterio de parada del tiempo de computación** se produce, para ambas instancias, **a partir de los 6 parámetros QoS**, y también para **valores muy altos del factor de división  $div$**  (más de 50).
- Los **criterios de parada antes de llegar a la cuarta generación**, por **exceso de precisión** o por **ausencia de soluciones** encontradas, suele darse para **valores altos del factor de división  $div$**  (a partir de 20).

En general, estas observaciones nos llevan a **considerar valores bajos del factor de división  $div$**  (3 o 5).

### **6.8.3.3 Situaciones en las que SEFISA mejora a SEFI**

El análisis de rendimiento SEFI frente a SEFISA es imprescindible para valorar la adecuación de la heurística adaptativa SEFISA, es decir, que la heurística que se ha desarrollado cumpla las expectativas que se apuntaron en la sección 6.6.5.

En la Figura 60, Figura 61 y Figura 62 se muestran los rendimientos de SEFISA frente a SEFI, entendido este rendimiento como el mejor valor de *fitness* encontrado (solución óptima encontrada) en relación al número total de combinaciones generadas por ambos algoritmos (*total combinations*) y al número de soluciones posibles (*valid combinations*), como gráficas separadas. Cada fila de gráficas corresponde a un número distinto de parámetros QoS (3, 4 o 5).

En la Figura 60 se muestran los resultados obtenidos con la instancia #1 para el perfil de usuario #1, la Figura 61 los muestra para el perfil de usuario #2 y la Figura 62 para el perfil #3.

Los puntos marcados en la línea SEFISA corresponden con los valores de *fitness* óptimos obtenidos en aquellas ejecuciones de SEFISA que tenían valores del factor de división *div* iguales a (contando desde el primero a la izquierda) 3, 5, 10, 20, 50 y 100. De estos puntos se han señalado, mediante unos círculos grises, aquellos valores que corresponden con factores de división  $div=3$  y  $div=5$ , y que representan un mejor comportamiento de SEFISA frente a SEFI.

Observando estas gráficas se puede concluir que, en general, **SEFISA obtiene un rendimiento igual o superior a SEFI para valores bajos del factor de división.**

La mejoría de SEFISA frente a SEFI se constata en los perfiles de usuario #1 y #3. Para el perfil de usuario #2, el rendimiento es peor en SEFISA en relación al número de combinaciones generadas, pero esto no interesa, porque lo importante es el hallazgo de combinaciones válidas, esto es, posibles soluciones. Por tanto, en este caso, SEFISA obtiene un rendimiento similar a SEFI.

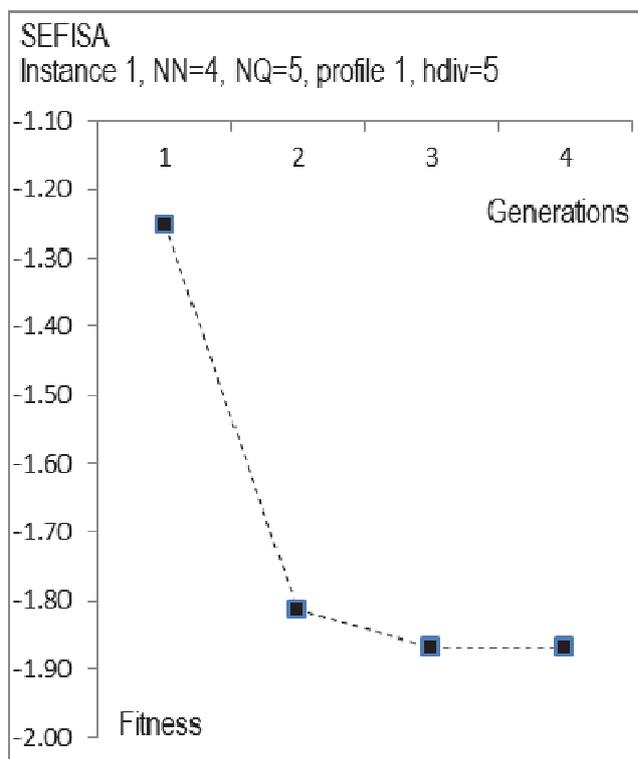


Figura 59. Gráfica representativa de la evolución de la solución óptima encontrada a lo largo de las generaciones de SEFISA. Datos obtenidos para la instancia #1 (4 redes inalámbricas de acceso) con 5 parámetros de calidad, para un grado de división de los espacios de búsqueda en SEFISA fijado en 5.

El hecho de utilizar valores bajos del factor de división, tal como se ha apuntado anteriormente, ofrece dos ventajas:

- Poder cumplir el requisito de tiempo real (tiempo de ejecución inferior a 1s), ya que los tiempos de computación aumentan notablemente con el tamaño de la variable *div*. Esto permite una ventaja adicional: poder trabajar con un mayor número de parámetros de calidad (que elevaban el tiempo de computación con valores altos de *div*).
- Poder evolucionar con más generaciones, facilitando la obtención de mejores valores de *fitness* y por tanto, soluciones más óptimas.

Se han realizado experimentos similares con la instancia experimental #2, obteniendo conclusiones análogas.

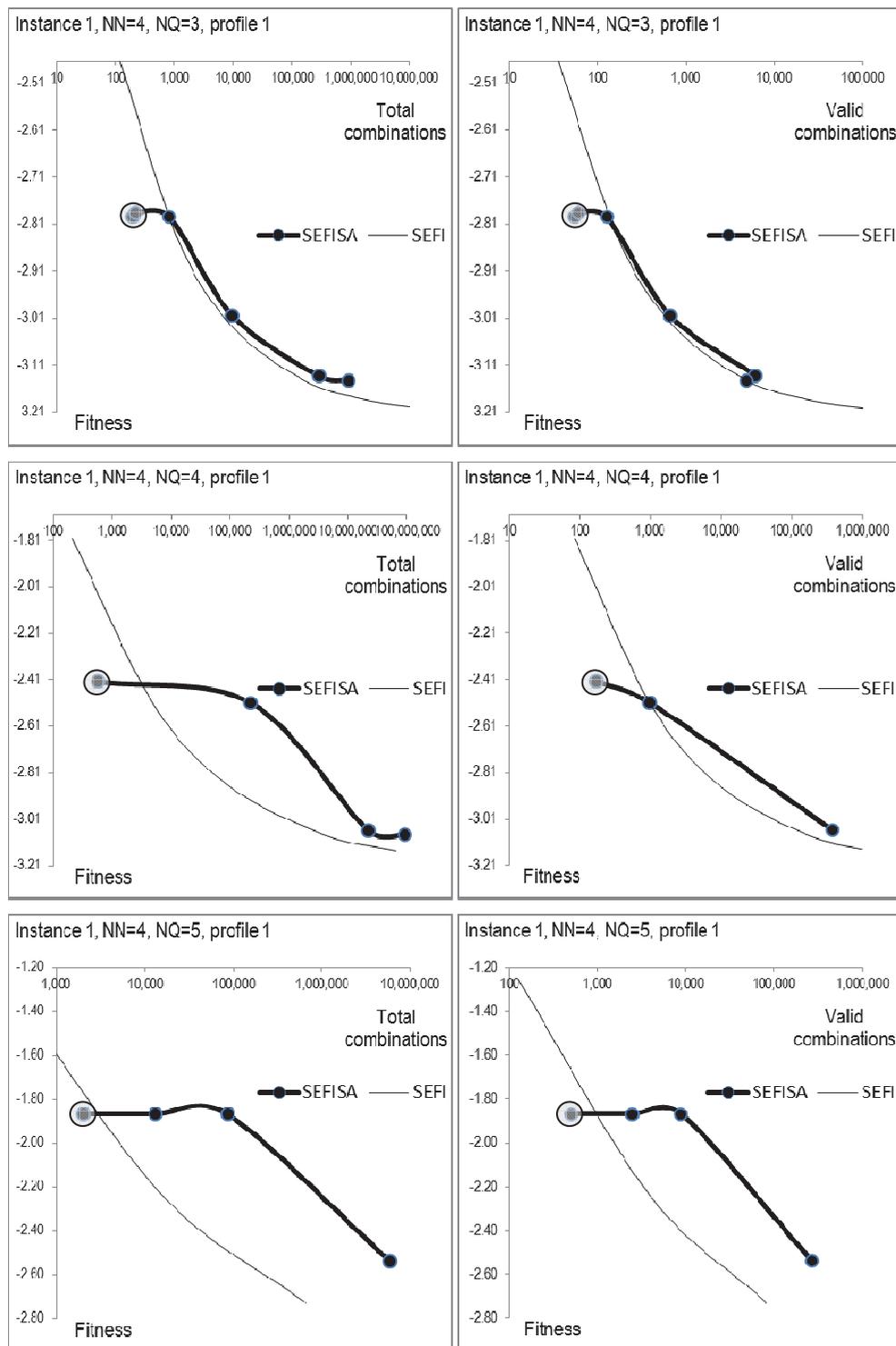


Figura 60. Rendimiento de SEFISA frente a SEFI, entendido como mejor *fitness* (solución) encontrado en relación al número total de combinaciones (*total combinations*) y al número de soluciones posibles (*valid combinations*), como gráficas separadas, para la instancia #1 y perfil #1. Cada fila de gráficas corresponde a un número distinto de parámetros de calidad. Los puntos marcados en la línea SEFISA corresponden con los *fitness* óptimos obtenidos en aquellas ejecuciones con valores del factor de división *div* iguales a (contando desde el primero a la izquierda) 3, 5, 10, 20, 50 y 100.

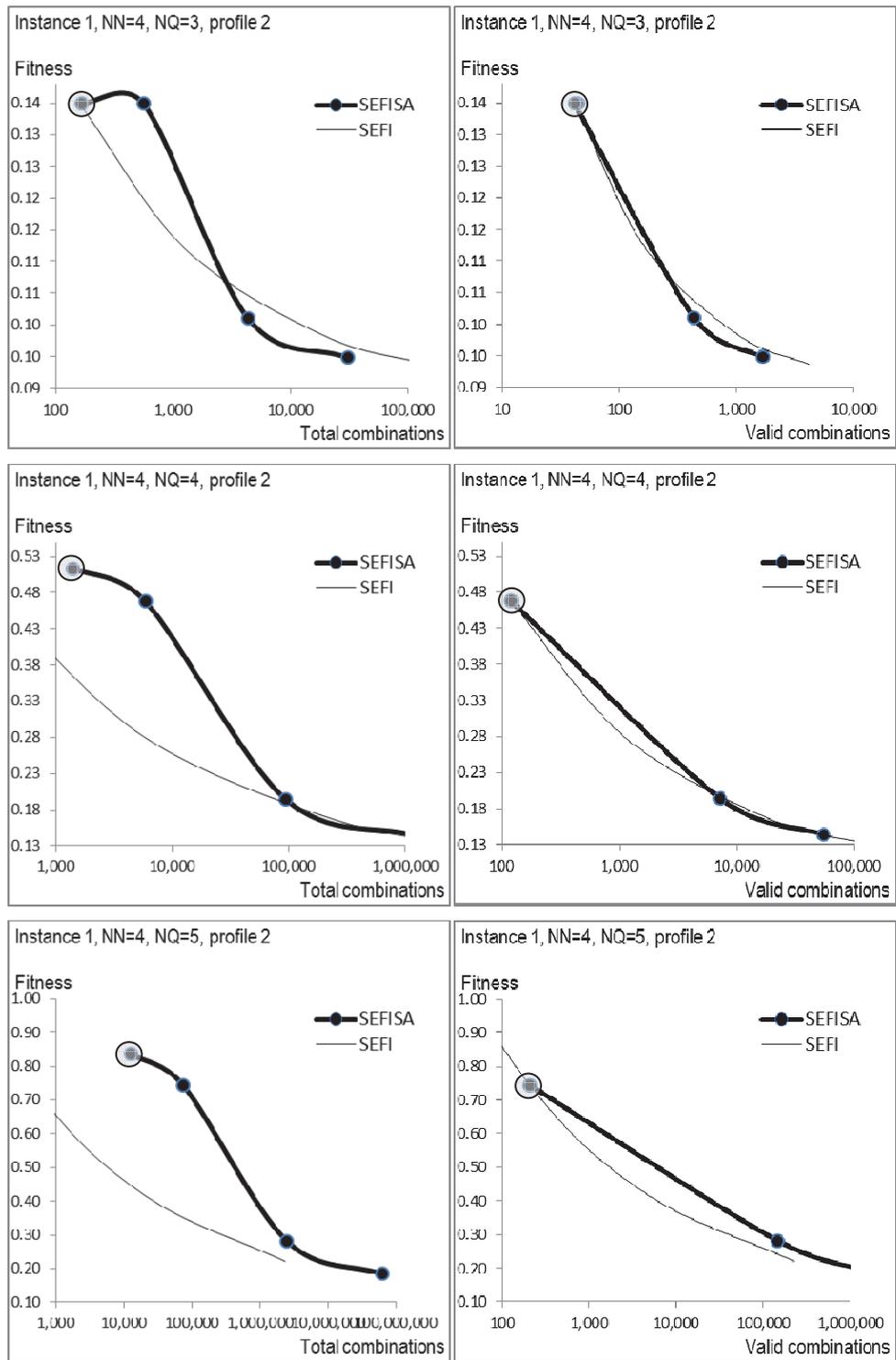


Figura 61. Los mismos gráficos mostrados en la Figura 60, pero para el perfil #2.

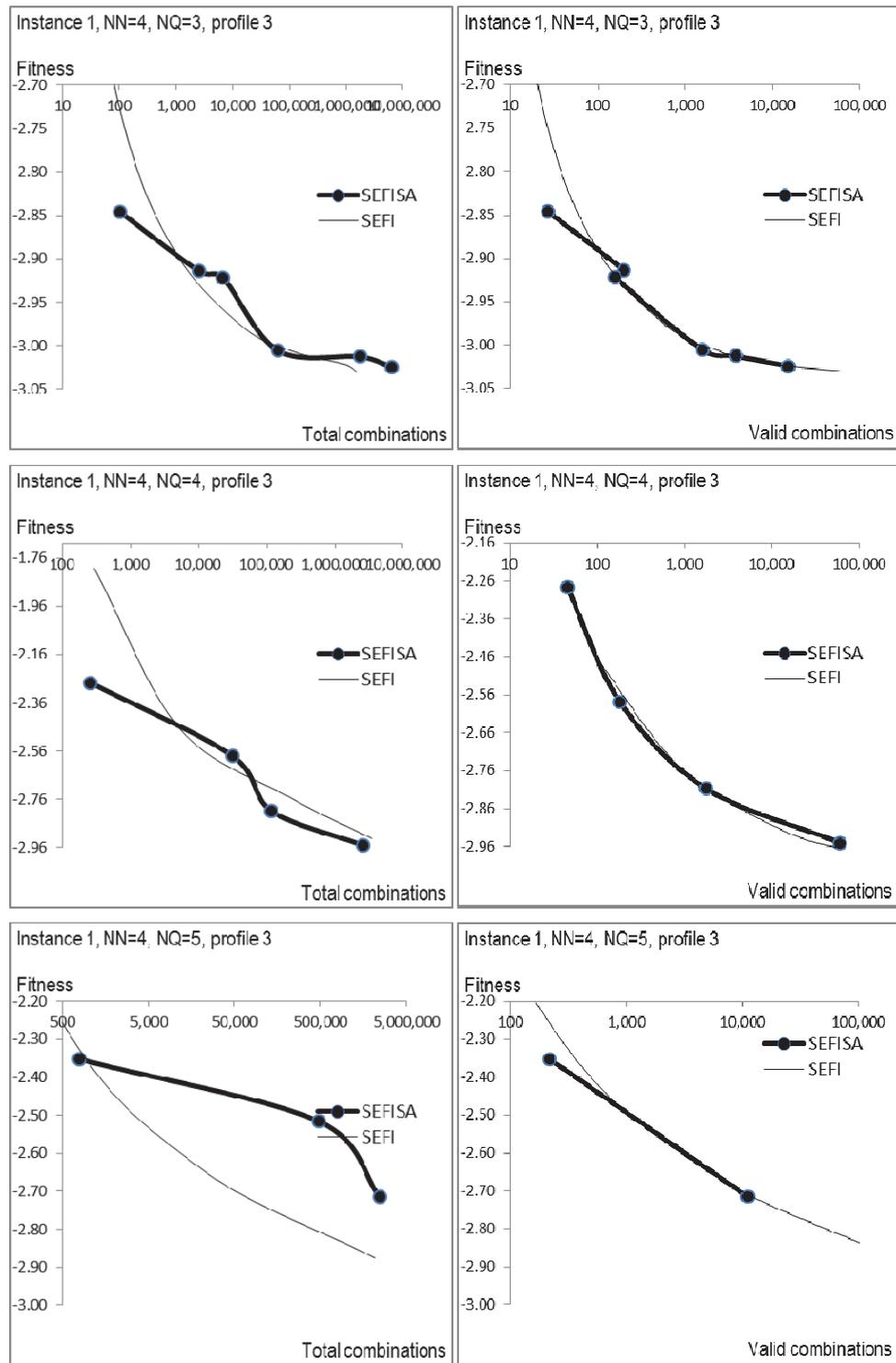


Figura 62. Los mismos gráficos mostrados en la Figura 60, pero para el perfil #3.

#### 6.8.3.4 Análisis de los resultados

Del análisis de las distintas gráficas y de los datos que se tienen tabulados en las bases de datos de los experimentos, se han podido constatar otras observaciones:

- A veces ocurre que una generación de SEFISA se ejecuta en menos tiempo que la generación precedente (cosa que no es lo general, ya que la reducción de los sucesivos espacios de búsqueda que son divididos por el mismo número produce una precisión de búsqueda menor). La razón se puede encontrar en el hecho de que hay casos donde, al pasar a una nueva generación, hay que recortar varios espacios de búsqueda de forma añadida a su propia reducción a la mitad, ya que exceden por algún lado los límites mínimos o máximos posibles (por ejemplo, ver el caso de la Figura 58). Este truncamiento añadido de los espacios de búsqueda hace que el número de valores de pesos para generar combinaciones sea mucho menor. Por ejemplo, en la instancia #2, NQoS=4 y hdiv=50, en la segunda generación se han generado 17 millones de combinaciones, mientras que en la primera fueron 57 millones.
- La primera generación de SEFISA, como indica la lógica, produce siempre los mismos resultados que una ejecución de SEFI con el valor de precisión de búsqueda  $h$  ajustado a lo que sería el  $h$  obtenido en SEFISA con el valor del factor de división  $div$  determinado.
- Hay valores del factor de división que producen prácticamente siempre los mismos resultados que los de otros valores, por lo que no se consideran como opción en la configuración o personalización de SEFISA. Por ejemplo,  $div=15$  da el mismo resultado que  $div=20$ , y  $div=25$  da el mismo resultado que  $div=50$ .

Se resumen a continuación las principales conclusiones derivadas del análisis de los resultados experimentales expuestos anteriormente.

- El tiempo de computación de SEFISA aumenta con el número de parámetros QoS y con el grado de precisión de la división de los espacios de búsqueda.
- Existe una tendencia general a encontrar una mejor solución óptima a lo largo de las generaciones de SEFISA. Esto demuestra el carácter evolutivo de la heurística que, basada en *Simulated Annealing*, fundamenta su evolución en la adaptación de los espacios de búsqueda y en la búsqueda directa de las soluciones en dichos espacios mediante la técnica exacta SEFI.

No obstante, a veces se estanca el hallazgo de mejores soluciones a lo largo de las sucesivas generaciones, cosa que no suele producirse si se utilizan valores bajos del factor de división.

- Los criterios de parada suelen producirse a partir de 6 parámetros de calidad, y también para valores muy altos del factor de división.
- En general, **SEFISA ofrece un rendimiento igual o superior a SEFI para valores bajos del factor de división**. Estos valores, además, aportan las ventajas de cumplir el requisito de tiempos de computación muy bajos, poder trabajar con más parámetros QoS y poder evolucionar con más generaciones, facilitando de esta manera la obtención de mejores soluciones.

#### **6.8.4 Comparativa de rendimiento con otros autores**

Para establecer una comparativa de rendimientos entre las heurísticas SEFI y SEFISA desarrolladas en esta tesis con otras técnicas, se necesita un mismo conjunto de datos experimentales y condiciones de entrada análogas para poder obtener resultados que puedan ser comparables de forma realista.

##### **6.8.4.1 Estudio de un caso real**

Los artículos que se han encontrado en revistas y congresos no muestran o publican las instancias experimentales, dificultando de esta manera la evaluación del rendimiento de SEFI y SEFISA. No obstante, se ha encontrado un trabajo que publica con suficiente detalle los datos experimentales. De hecho, se han utilizado esos datos para configurar una de las instancias experimentales utilizadas en esta tesis, con el propósito de hacer la comparativa de rendimientos.

Así, en [Lassoued 2008] se muestra un trabajo de optimización de pesos asignados a parámetros de calidad para la misma instancia #2 que se ha considerado en todos los experimentos con SEFI y SEFISA. En dicho trabajo, sus autores identifican los parámetros QoS de la siguiente manera:

- B = Bandwidth.
- E = BER.
- D = Average delay.
- S = Security level.
- C = Monetary cost.

Los autores del trabajo sostienen que los valores de los pesos deben tener en cuenta los requisitos de las aplicaciones que se ejecutan en el terminal móvil. Así, el parámetro más importante para conversación es el retardo, mientras que para flujos de *streaming* se requieren grandes anchos de banda y retardos aceptables. De acuerdo con estos requisitos, los autores de este trabajo comparan los diferentes parámetros en función del peso de importancia que les asignan, tal como se muestra en la Figura 63.

En dicho trabajo se evalúan y cuantifican los pesos mediante el ya citado algoritmo **AHP** [Song 2005], con el cual los autores obtienen una combinación óptima (ellos llaman combinación “adecuada”) para cada tipo de servicio, flujo de aplicación o perfil de usuario, y que se muestra en la Tabla 19. Nótese que la suma de los pesos de la primera combinación no suma exactamente 1, por lo que, a efectos de comparativa con las heurísticas desarrolladas en la tesis, se decidió utilizar el peso asignado a D como 0,614 en lugar de 0,613.

ATTRIBUTES COMPARISON FOR CONVERSATIONAL FLOW						ATTRIBUTES COMPARISON FOR STREAMING FLOW					
	B	E	D	S	C		B	E	D	S	C
B	Equally Important	-	-	-	-	B	Equally Important	-	-	-	-
E	Equally Important	Equally Important	-	-	-	E	Very Less Important	Equally Important	-	-	-
D	Very More Important	Very More Important	Equally Important	-	-	D	Moderately Less Important	Moderately More Important	Equally Important	-	-
S	Slightly More Important	Slightly More Important	Moderately Less Important	Equally Important	-	S	Moderately Less Important	Moderately More Important	Slightly Less Important	Equally Important	-
C	Slightly More Important	Slightly More Important	Moderately Less Important	Equally Important	Equally Important	C	Moderately Less Important	Moderately More Important	Slightly Less Important	Equally Important	Equally Important

Figura 63. Comparativas de atribuciones de importancia entre los distintos parámetros QoS de la instancia #2 en [Lassoued 2008], según el tipo de aplicación requerida por el usuario (conversación o transmisión multimedia).

Service	B	E	D	S	C	Sum
Conversational	0.065	0.065	0.613	0.128	0.128	0.999
Streaming	0.545	0.035	0.178	0.121	0.121	1

Tabla 19. Pesos óptimos asignados a los distintos parámetros QoS para dos tipos de servicio, según la instancia #2 en el trabajo [Lassoued 2008].

La metodología que utilizan estos autores permite a un terminal móvil escoger dinámicamente la red de acceso más adecuada para cada aplicación. Por ejemplo, en la Figura 64 se muestra el historial de la selección de la mejor red a la que se puede conectar un terminal móvil durante su trayectoria, teniendo en cuenta los dos tipos de aplicación. Así por ejemplo, las aplicaciones de conversación son a menudo transmitidas a través de la red de acceso que ofrece el retardo más bajo (N5) mientras que las aplicaciones de *streaming* son transmitidas a través de la red WiMax que ofrece el ancho de banda más grande y un retardo aceptable. De esta forma, el software que ejecuta el VH escoge la red de acuerdo al tipo de aplicación, las preferencias de usuario y los rendimientos de cada red de acceso.

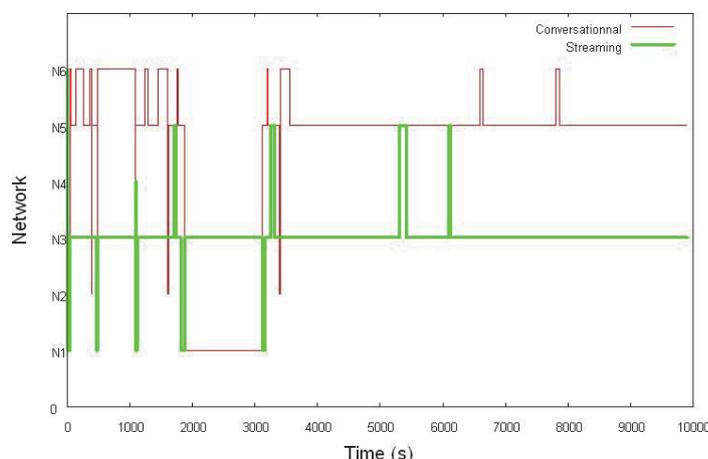


Figura 64. Selección de la mejor red en cada instante de la trayectoria de un terminal móvil que se desplaza descubriendo distintas redes inalámbricas heterogéneas, teniendo en cuenta los dos posibles tipos de aplicación.

### 6.8.4.2 Análisis frente a SEFI

Dado que la heurística SEFISA se basa en SEFI y, como se ha visto anteriormente, iguala o mejora los resultados de SEFI en general, **se ha efectuado la comparativa de rendimientos únicamente con los resultados de SEFI.**

En la Tabla 20 se muestran los resultados obtenidos por los autores de este artículo y los resultados que se han obtenido en esta tesis a partir de SEFI. La tabla muestra las combinaciones de pesos que satisfacen la restricción (soluciones), para la instancia experimental #2 (6 redes inalámbricas y 5 parámetros de calidad de servicio).

	Solución (pesos)					Red óptima	Fitness
	w0	w1	w2	w3	w4		
	B	E	D	S	C		
[Lassoued 2008] (perfil #2)	0.065	0.065	0.614	0.128	0.128	4	1.25963136
[Lassoued 2008] (perfil #3)	0.545	0.035	0.178	0.121	0.121	4	-4.428212445
SEFI (h=0.01, perfil #2)	0.47	0.01	0.5	0.01	0.01	4	-2.08185
SEFI (h=0.005, perfil #2)	0.485	0.005	0.5	0.005	0.005	4	-2.17334
SEFI (h=0.01, perfil #3)	0.6	0.28	0.1	0.01	0.01	4	-6.03799
SEFI (h=0.005, perfil #3)	0.6	0.29	0.1	0.005	0.005	4	-6.0705

Tabla 20. Combinaciones de pesos que satisfacen la restricción (soluciones), para la instancia experimental #2 (6 redes inalámbricas, 5 parámetros QoS). Las soluciones aportadas por [Lassoued 2008] han sido calculadas mediante el algoritmo AHP, para el perfil #2 (conversación, donde importa el retardo  $D$ :  $0.5 < D < 0.7$ ) y para el perfil #3 (*streaming*, donde importa un gran ancho de banda  $B$  y un retardo aceptable  $D$ :  $0.5 < B < 0.7$ ,  $0.1 < D < 0.3$ ). Las soluciones encontradas por SEFI tienen mucho mejor fitness en todos los casos, aún con baja precisión de búsqueda.

Las soluciones aportadas por [Lassoued 2008] han sido calculadas mediante el algoritmo AHP, para dos perfiles de usuario:

- Perfil #2: conversación. Importa el retardo  $D$ , con los límites  $0,5 < D < 0,7$ .
- Perfil #3: streaming. Importa un gran ancho de banda  $B$  y un retardo aceptable  $D$ , con los límites  $0,5 < B < 0,7$  y  $0,1 < D < 0,3$ .

### 6.8.5 Conclusiones

Del análisis de los resultados anteriores se puede concluir que **las soluciones encontradas por SEFI tienen mucho mejor fitness en todos los casos, aún con una baja precisión de búsqueda.**

Si bien la red inalámbrica óptima encontrada por AHP y por SEFI es la misma, en otras situaciones, tal como se ha demostrado en la sección 6.6.3, encontrar un fitness mucho mejor puede implicar cambiar a otra red distinta, por lo que el algoritmo SEFI es mucho más adecuado que AHP para la fase de decisión VH.

## 6.9 Algoritmo GAVH

En este apartado se explica una propuesta de Algoritmo Genético para la fase de decisión VH, **GAVH**. Este algoritmo ha sido desarrollado a partir de las motivaciones expuestas en la sección 6.6.5: idear heurísticas de selección óptima de redes en VH que se muevan en entornos bajos de precisión para poder encontrar buenas soluciones en tiempos muy pequeños.

Si bien el algoritmo SEFISA ya satisface plenamente dicho objetivo de encontrar buenas soluciones en tiempos pequeños, como ha visto anteriormente, el planteamiento de una heurística basada en algoritmos genéticos añade más perspectivas de resolución robusta del problema, al evitar ciertos problemas de estancamiento de soluciones en SEFISA, tal como se apuntó en el epígrafe “otras consideraciones” de la sección 6.8.3., y al proseguir en la línea de obtención de buenas soluciones de forma muy rápida.

### 6.9.1 *Características*

El algoritmo GAVH maneja poblaciones donde los individuos son, al igual que en el caso del algoritmo SEFISA, posibles soluciones del problema de optimización, es decir, vectores de pesos QoS que cumplen la restricción, lo cual permitirá comparativas de rendimientos entre ambas heurísticas.

En la Figura 65 puede observarse cómo se plantea la población, que será generada inicialmente con un número par de individuos. Cada individuo es identificado por la variable  $w\_indiv[ip][iq]$ , donde  $ip$  es un índice correspondiente al número del individuo en la población, e  $iq$  es un índice correspondiente al parámetro QoS.

A su vez, el *fitness* correspondiente a cada individuo es, realmente, un vector de valores de *fitness*, ya que existen varias redes de distinta tecnología. Por tanto, el *fitness* de un individuo es identificado por la variable  $F[ip][in]$ , donde  $ip$  es un índice que corresponde con la red inalámbrica.

Finalmente, se considera que las poblaciones, a lo largo de las generaciones, tienen un tamaño fijo, determinado por la constante *popsize*.

A la hora de generar la población inicial, el algoritmo necesita fijar valores a las siguientes variables:

- $NQ$  = número de parámetros QoS.
- $NN$  = número de redes inalámbricas.
- $h$  = intervalo de generación de pesos, entre 0 y 1.
- *popsize* = tamaño de la población.

Para la generación de la población inicial se utilizará el algoritmo de búsqueda exhaustiva SEFI. Si el tamaño de la población no es demasiado grande, la generación de la población inicial con el algoritmo SEFI es muy rápida, incluso para un elevado número de parámetros QoS.

Utilizar SEFI para la generación de la población inicial es una estrategia muy útil frente a una política clásica de generación aleatoria, puesto que de esta manera la selección de vectores de pesos cubre de manera uniforme todo el espacio de soluciones, según el intervalo de búsqueda  $h$  fijado. Esto permitirá que el algoritmo genético parta de una población inicial donde los individuos representan todas las zonas del espacio global de soluciones, permitiendo un elevado grado de diversidad para las sucesivas evoluciones de la población.

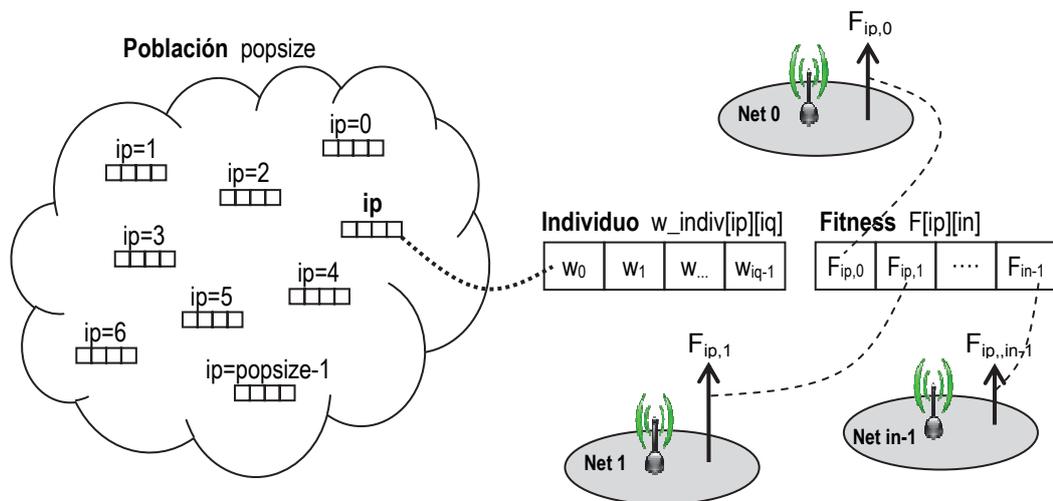


Figura 65. Planteamiento de la población, individuos y valores de *fitness* para el algoritmo genético GAVH.

Por ejemplo, en la generación de una población inicial para 3 redes inalámbricas, 4 parámetros QoS, un intervalo de búsqueda de 0.01 y tamaño de población de 100 individuos, el algoritmo SEFI genera los siguientes individuos para la población inicial ( $NN=3$ ,  $NQ=4$ ,  $h=0.01$ ,  $popsiz=100$ ):

```
w_indiv[0][iq]={0.01,0.01,0.01,0.97,} --- fitness_indiv[0][in]={-0.0560894,-2.19767,-1.51972,}
w_indiv[1][iq]={0.01,0.18,0.56,0.25,} --- fitness_indiv[1][in]={1.7288,1.90905,2.11892,}
w_indiv[2][iq]={0.01,0.41,0.37,0.21,} --- fitness_indiv[2][in]={1.99276,2.1372,2.42571,}
w_indiv[3][iq]={0.01,0.88,0.07,0.04,} --- fitness_indiv[3][in]={2.73539,3.10685,3.48878,}
w_indiv[4][iq]={0.02,0.18,0.24,0.56,} --- fitness_indiv[4][in]={0.987718,0.0746777,0.523728,}
w_indiv[5][iq]={0.02,0.41,0.28,0.29,} --- fitness_indiv[5][in]={1.78128,1.6147,1.9671,}
w_indiv[6][iq]={0.02,0.96,0.01,0.01,} --- fitness_indiv[6][in]={2.83697,3.21176,3.61171,}
[.....]
w_indiv[96][iq]={0.65,0.08,0.15,0.12,} --- fitness_indiv[96][in]={0.207587,-1.5863,-1.33599,}
w_indiv[97][iq]={0.68,0.08,0.19,0.05,} --- fitness_indiv[97][in]={0.286932,-1.38564,-1.17992,}
w_indiv[98][iq]={0.72,0.04,0.17,0.07,} --- fitness_indiv[98][in]={0.101878,-1.76452,-1.55085,}
w_indiv[99][iq]={0.77,0.1,0.02,0.11,} --- fitness_indiv[99][in]={-0.0817851,-2.32368,-2.03644,}
```

En la Figura 66 se resumen las tres principales operaciones que se ejecutan en una generación de GAVH.

La primera generación comienza tras definir la población inicial tal como se acaba de ver, y una vez completada da paso a nuevas generaciones que se suceden cíclicamente. A continuación se describen estas operaciones.

### 6.9.1.1 Selección

La fase de selección se realiza en los siguientes pasos sucesivos:

- Clasificar los individuos por su valor de *fitness* (mejor individuo cuando menor sea el valor de su función de *fitness*).
- Seleccionar la mitad de la población ( $popsiz/2$ ) consistente en los individuos que tengan los mejores valores de *fitness*. Estos individuos serán los llamados “progenitores A”, y sobrevivirán a esta generación.
- Seleccionar la mitad de la población consistente en los individuos que tengan los peores valores de *fitness*. Estos individuos serán los “progenitores B”, y no sobrevivirán a esta generación.

Se considera como una restricción del algoritmo GAVH que la población inicial tenga un número par de individuos ( $popsiz$ ), para de esta forma garantizar el mismo número de individuos en ambos grupos de progenitores.

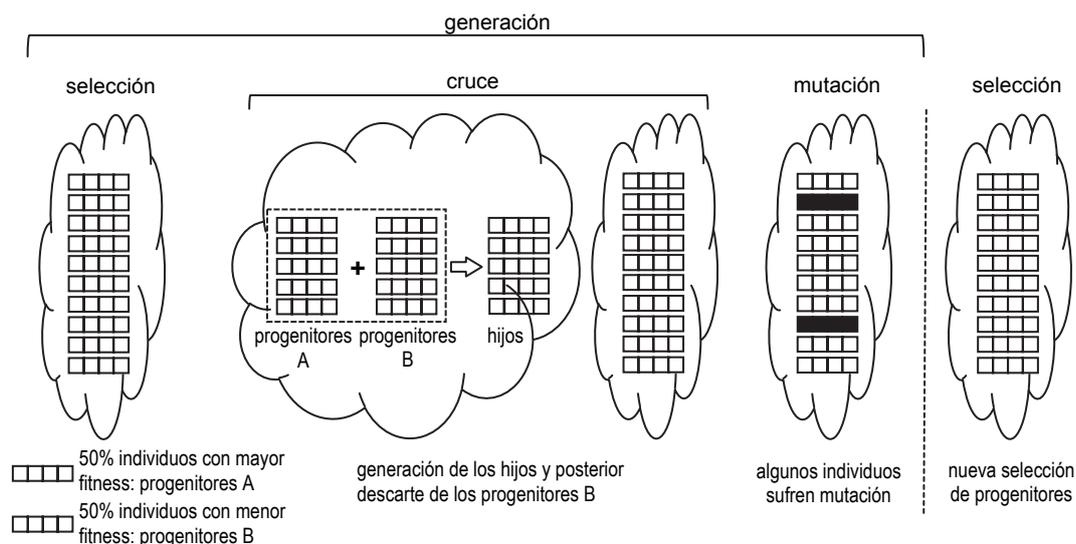


Figura 66. Operaciones del algoritmo genético que se suceden en una generación.

### 6.9.1.2 Cruce

En la fase de cruce (Figura 67), a cada progenitor A ( $wA$ ) le corresponde un progenitor B ( $wB$ ), formando una pareja única, cuyos miembros no serán utilizados en otras parejas. Los genes del hijo  $wS$  se generan calculando la media aritmética entre los dos genes correspondientes de los progenitores, según (42).

No obstante, cumplir la restricción impuesta en (27) implica que esta media aritmética pueda no aplicarse a partir de algún gen, en cuyo caso los genes restantes toman un valor equitativo (43), de forma que la suma final sea 1.

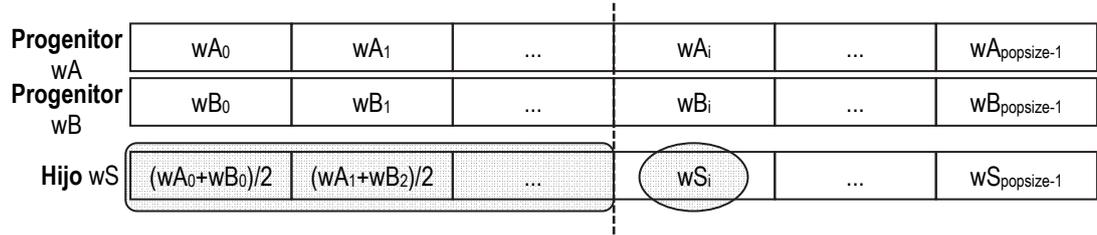


Figura 67. Los progenitores wA y wB generan el hijo wS, donde wS<sub>i</sub> es el gen o peso QoS i-ésimo del individuo. Este gen se genera de acuerdo con las reglas expresadas en las ecuaciones (42) y (43)217.

$$(42) \quad wS_i = \frac{wA_i + wB_i}{2}, \text{ si } \left( \frac{wA_i + wB_i}{2} + \sum_{j=0}^{i-1} \frac{wA_j + wB_j}{2} \right) < 1$$

$$(43) \quad wS_k = R, \text{ si } \left( \frac{wA_i + wB_i}{2} + \sum_{j=0}^{i-1} \frac{wA_j + wB_j}{2} \right) \geq 1, \text{ donde } R = \frac{1 - \left( \sum_{j=0}^{i-1} \frac{wA_j + wB_j}{2} \right)}{\text{popsize} - i}, \forall k = \{i, \dots, \text{popsize} - 1\}$$

### 6.9.1.3 Mutación

La operación de cruce anterior es una operación exacta, es decir, libre de aleatoriedad. Esto supone que varias ejecuciones del mismo experimento con el algoritmo genético darán siempre el mismo resultado.

Con el fin de que se pueda introducir cierto grado de diversidad a la población, es común introducir en los AGs algunas políticas de mutación que le otorgan, de paso, un carácter aleatorio que requerirá de la realización de varias ejecuciones y del uso de técnicas estadísticas para generar y analizar los resultados.

En este trabajo se consideran dos propuestas de políticas de mutación, identificadas por la variable *smut* (cuando no hay mutación se considera *smut*=0):

- Mutación por desplazamiento (*smut*=1). Esta política de mutación desplaza en *qmut* posiciones hacia la derecha los genes del cromosoma del individuo. Es decir, los pesos QoS que conforman el individuo *w* se desplazan hacia la derecha en una cuantía determinada, tal como se observa en la Figura 68. Obviamente, si *qmut*=0 no hay mutación, al no haber desplazamiento. El valor máximo de *qmut* sería de *NQ*-1, y se fija al principio de cada ejecución del algoritmo genético.
- Mutación por intercambio (*smut*=2). Esta política de mutación intercambia dos genes del individuo, según dos posiciones predefinidas. Para definir estas dos posiciones se pueden seguir distintas políticas seleccionadas por el valor *qmut*:
  - Si *qmut*=0, las dos posiciones se escogen aleatoriamente al comienzo de cada ejecución, siendo por tanto las mismas para todos los individuos de la población en cualquier generación, pero distintas en cada ejecución.

- Si  $qmut=1$ , las dos posiciones se escogen aleatoriamente al comienzo de cada generación, siendo las mismas para todos los individuos de la población en la misma generación, pero distintas en otras generaciones y ejecuciones del AG.
- Si  $qmut=2$ , las dos posiciones se escogen aleatoriamente para cada individuo al comienzo de cada generación, pudiendo ser, por tanto, distintas para cada individuo, generación y ejecución del AG.

Los individuos de la población que serán mutados son seleccionados de forma aleatoria en cada generación. Por otro lado, el porcentaje de estos individuos de la población que sufrirán mutación,  $pmut$  (%), es un parámetro que se fija en cada experimento (es decir, es el mismo para todas las ejecuciones del experimento), y se procura que no sea muy alto.

La aleatoriedad en la selección de individuo o posiciones de los genes se implementa mediante la función *rand* en C, asegurando una aleatoriedad real (usando la función *srand*, que se vincula al instante de ejecución del algoritmo con la función *time*).

## **6.9.2 Implementación**

Se han considerado los siguientes factores a la hora de implementar GAVH:

### **6.9.2.1 Programación del algoritmo**

El algoritmo ha sido enteramente programado en lenguaje ANSI C, con el propósito de hacerlo portable a cualquier entorno de procesamiento existente en los diferentes dispositivos de comunicaciones móviles donde se ejecutaría. No se ha utilizado ninguna librería externa o preexistente: toda la arquitectura y políticas del algoritmo han sido programadas de forma específica para este problema.

### **6.9.2.2 Instancias experimentales**

Con el fin de comprobar el funcionamiento y validez del algoritmo, así como para evaluar su rendimiento al comparar sus resultados con los de SEFI y SEFISA, se ha optado por utilizar las mismas instancias experimentales en todos los casos.

### **6.9.2.3 Experimentos y ejecuciones**

Dado que este algoritmo tiene un cierto grado de impredecibilidad de los resultados debido a sus políticas de mutación (que comportan aleatoriedad), es necesario que cada experimento sea ejecutado más de una vez, con el fin de obtener conclusiones estadísticamente fiables. Con este propósito, se ha fijado en 50 el número de ejecuciones efectuadas para cada experimento.

Por otro lado, se han efectuado numerosos experimentos, consistentes en diferentes configuraciones tanto del algoritmo (tamaño de población, número de generaciones, política de mutación, valor de la mutación) como del problema (instancia experimental, perfil de usuario, número de redes inalámbricas, número de parámetros de calidad de servicio), con el propósito de sintonizar parámetros

(obtener los valores más apropiados para las diferentes variables del algoritmo mediante un barrido paramétrico), efectuar comparativas de precisión de los resultados y evaluar esfuerzos computacionales.

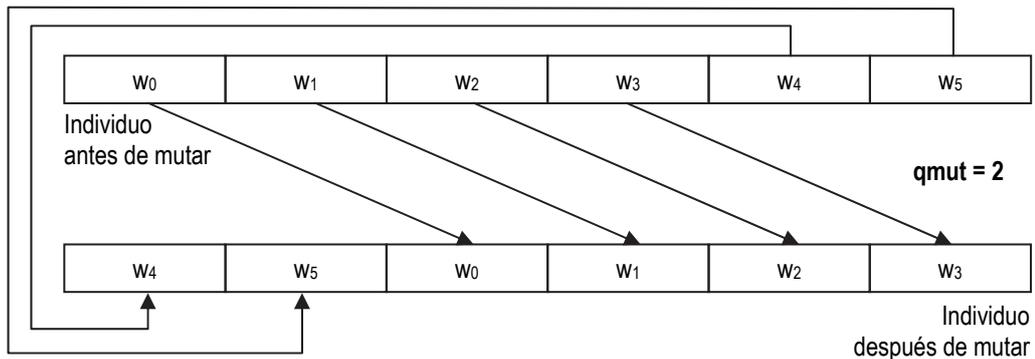


Figura 68. Mutación por desplazamiento ( $smut=1$ ). Los genes del cromosoma del individuo se desplazan en varias posiciones hacia la derecha (en el dibujo, dos posiciones:  $qmut=2$ ).

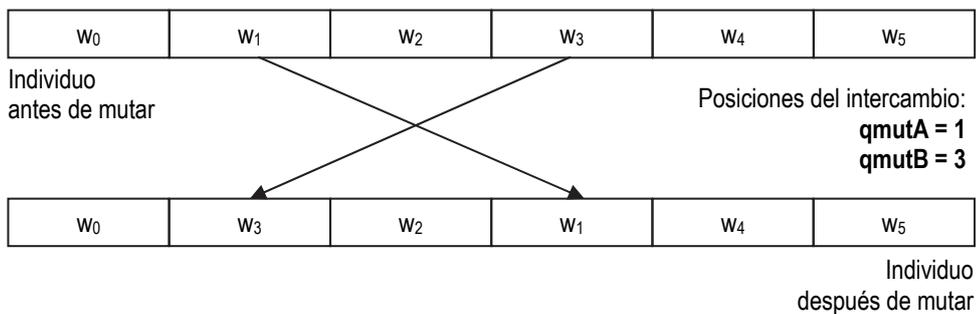


Figura 69. Mutación por intercambio ( $smut=2$ ). Dos genes del cromosoma del individuo intercambian sus posiciones entre sí. Estas posiciones pueden ser fijas en una misma ejecución (pero distintas en otras), en una misma generación (pero distintas en otras), o cambiar aleatoriamente para cada individuo en cada generación y en cada ejecución.

#### **6.9.2.4 Estadísticas para los resultados**

Cada experimento consta de 50 ejecuciones del algoritmo con la misma configuración, recogiendo como resultados los siguientes valores:

- *Fitness* mínimo. El menor de los 50 valores del *fitness*.
- *Fitness* máximo. El mayor de los 50 valores del *fitness*.
- *Fitness* promedio. Es la media aritmética de los 50 valores del *fitness*.
- Varianza. Informa de la dispersión de los resultados del *fitness*, calculándose como la media de las diferencias cuadráticas de los 50 valores de *fitness* con respecto a su media aritmética.
- Desviación estándar. Mide cuánto se separan los 50 valores de *fitness* obtenidos en el experimento, calculándose como la raíz cuadrada de la varianza.

A efectos de solución aportada por GAVH, se considerará el *fitness* promedio, aunque existan mejores valores reportados como *fitness* mínimo.

#### **6.9.3 Resultados experimentales**

Se han efectuado numerosos experimentos con el propósito de ajustar los principales parámetros del algoritmo genético, determinar las mejores políticas de mutación, evaluar el comportamiento general del algoritmo y medir su precisión respecto a otras soluciones heurísticas.

##### **6.9.3.1 Barrido paramétrico**

Este epígrafe explica los detalles del barrido paramétrico efectuado sobre el algoritmo para la selección de la política de mutación, tamaño de la población y número de generaciones.

Con el objeto de seleccionar los mejores valores de los principales parámetros de GAVH, se ha realizado un conjunto de experimentos que parten de una configuración común de la instancia experimental:

- *profile* = 1 (perfil de usuario, el más genérico).
- *InstanceID* = 1 (instancia de datos del escenario de redes inalámbricas).
- *NN* = 4 (número de redes).
- *NQ* = 4 (número de parámetros de calidad de servicio).
- 50 ejecuciones para cada experimento.

Los experimentos hacen variar los valores de los parámetros *NG* (número de generaciones como criterio de parada), *popsiz*e (tamaño de población inicial, que se mantiene constante a lo largo de las generaciones), *h* (precisión en la búsqueda de soluciones) y *smut* (política de mutación). Este último parámetro implica, a su vez, el ajuste de otros parámetros adicionales correspondientes a las distintas políticas de mutación: *pmut* (porcentaje de individuos de la población que sufrirán

la mutación) y  $qmut$  (posición del gen del cromosoma que será mutado y que reorganizará todo el cromosoma para que se cumpla la restricción).

Para este barrido paramétrico se han efectuado alrededor de **21.000 ejecuciones**, que corresponden a 4 tandas experimentales (combinaciones de los valores  $h = \{0,01; 0,005\}$  y  $pmut = \{5; 20\}$ ) x 15 clases de experimentos (combinaciones de los valores  $NG = \{50; 500; 2,000\}$  y  $popsiz = \{50; 100; 500; 1.000; 2.500\}$ ) x 7 experimentos ( $smut = 0$  -sin mutación-, y combinaciones de los valores  $qmut = \{1; 2; 3\}$  para  $smut=1$  y  $qmut = \{0; 1; 2\}$  para  $smut=2$ ) x 50 ejecuciones = 21.000 ejecuciones en total.

En la Figura 70 se muestra la evolución de la mejor solución encontrada por GAVH (*fitness* promedio de 50 ejecuciones) respecto al tamaño de la población, para el conjunto de experimentos del barrido paramétrico, donde varían  $NG$ ,  $popsiz$  y  $smut$  (con la correspondiente variación de  $qmut$ ), distinguiéndose en dicha figura cuatro gráficas, que corresponden con los siguientes casos:

- (a)  $h = 0,005$  y  $pmut = 5$ .
- (b)  $h = 0,01$  y  $pmut = 5$ .
- (c)  $h = 0,005$  y  $pmut = 20$ .
- (d)  $h = 0,01$  y  $pmut = 20$ .

Del análisis de estas gráficas se deduce que, en general, el mejor comportamiento se alcanza seleccionando un bajo porcentaje de individuos mutados, es decir, con  $pmut=5$  (un 5%), tal como se observa en las gráficas (a) y (b).

Por otro lado, comparando estas dos gráficas se constata también cómo la mejoría se produce con un valor más bajo de  $h$ , lo cual indica una mayor precisión en la generación y búsqueda de soluciones, cosa que es lógico como ya se apuntó anteriormente en varios lugares de esta memoria de tesis.

Por tanto, como primera conclusión, se considera que es mejor utilizar bajos porcentajes de mutación.

Para ajustar esta política de mutación, determinar el mejor tamaño de población y el número de generaciones más adecuado, se muestran los resultados de la Figura 70 diseminados en tres figuras distintas (sin mutación y con mutaciones #1 y #2) y ampliadas en su eje de ordenadas, para extraer mejor la información.

Así, la Figura 71 muestra únicamente los casos que no consideran mutación ( $smut=0$ ), la Figura 72 muestra los casos que consideran mutación  $smut=1$  y la Figura 73 cuando la política de mutación es  $smut=2$ .

Por otro lado, la Figura 71 muestra las desviaciones típicas de las soluciones óptimas encontradas para todos los casos en los que hay presente mutación.

Nótese que en la Figura 71, Figura 72, Figura 73 y Figura 74 los rótulos de los ejes de abscisas  $\{1; 2; 3; 4; 5\}$  indican los tamaños de población  $\{50; 100; 500; 1.000; 2.500\}$  (número de individuos).

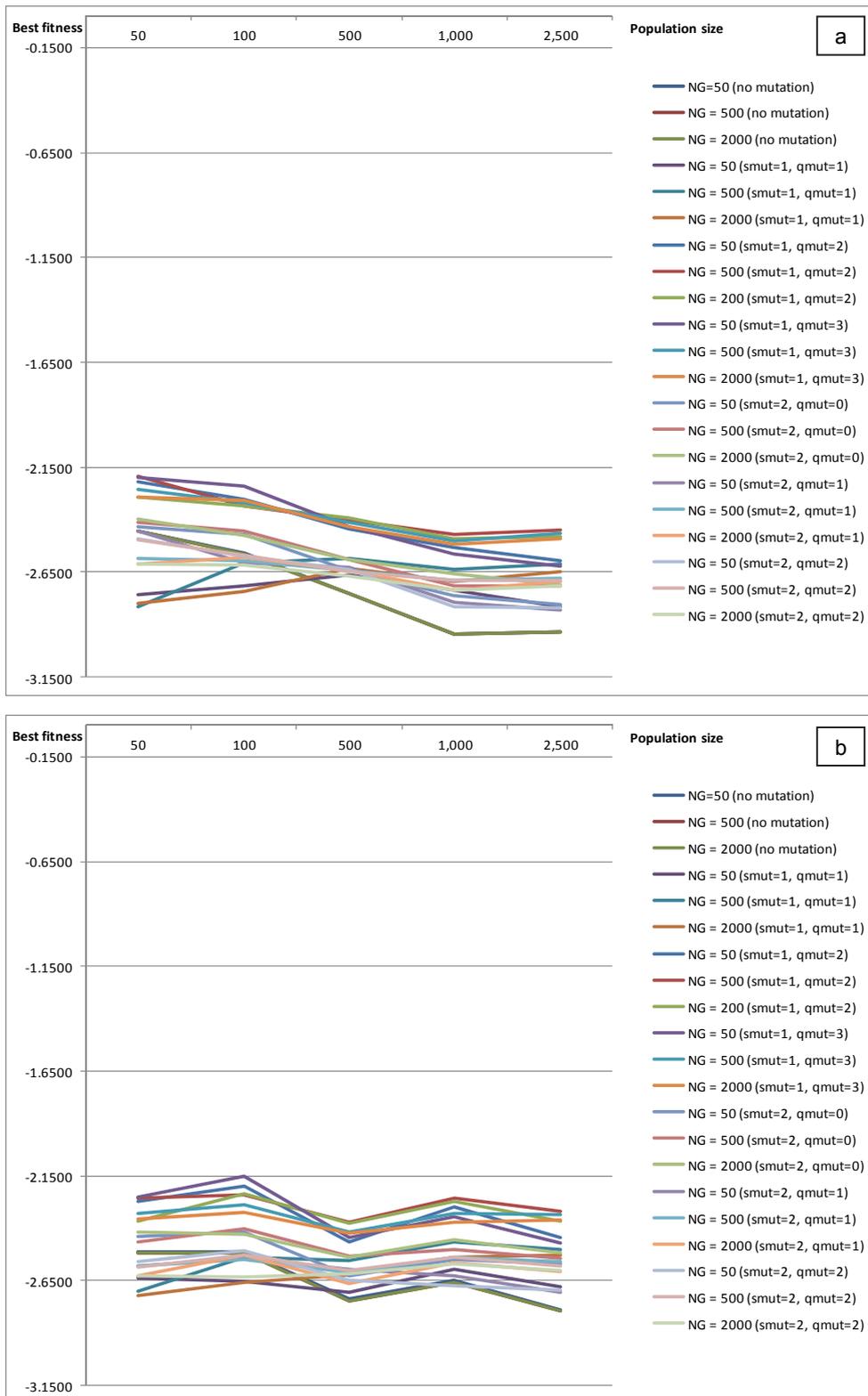


Figura 70. Evolución de la mejor solución encontrada por GAVH (*fitness* promedio de 50 ejecuciones) respecto al tamaño de la población, para el conjunto de experimentos del barrido paramétrico donde varían *NG*, *popsiz*e y *smut* (con la correspondiente variación de *qmut*): (a)  $h = 0,005$  y  $pmut = 5$ ; (b)  $h = 0,01$  y  $pmut = 5$ ; (c)  $h = 0,005$  y  $pmut = 20$ ; (d)  $h = 0,01$  y  $pmut = 20$ .

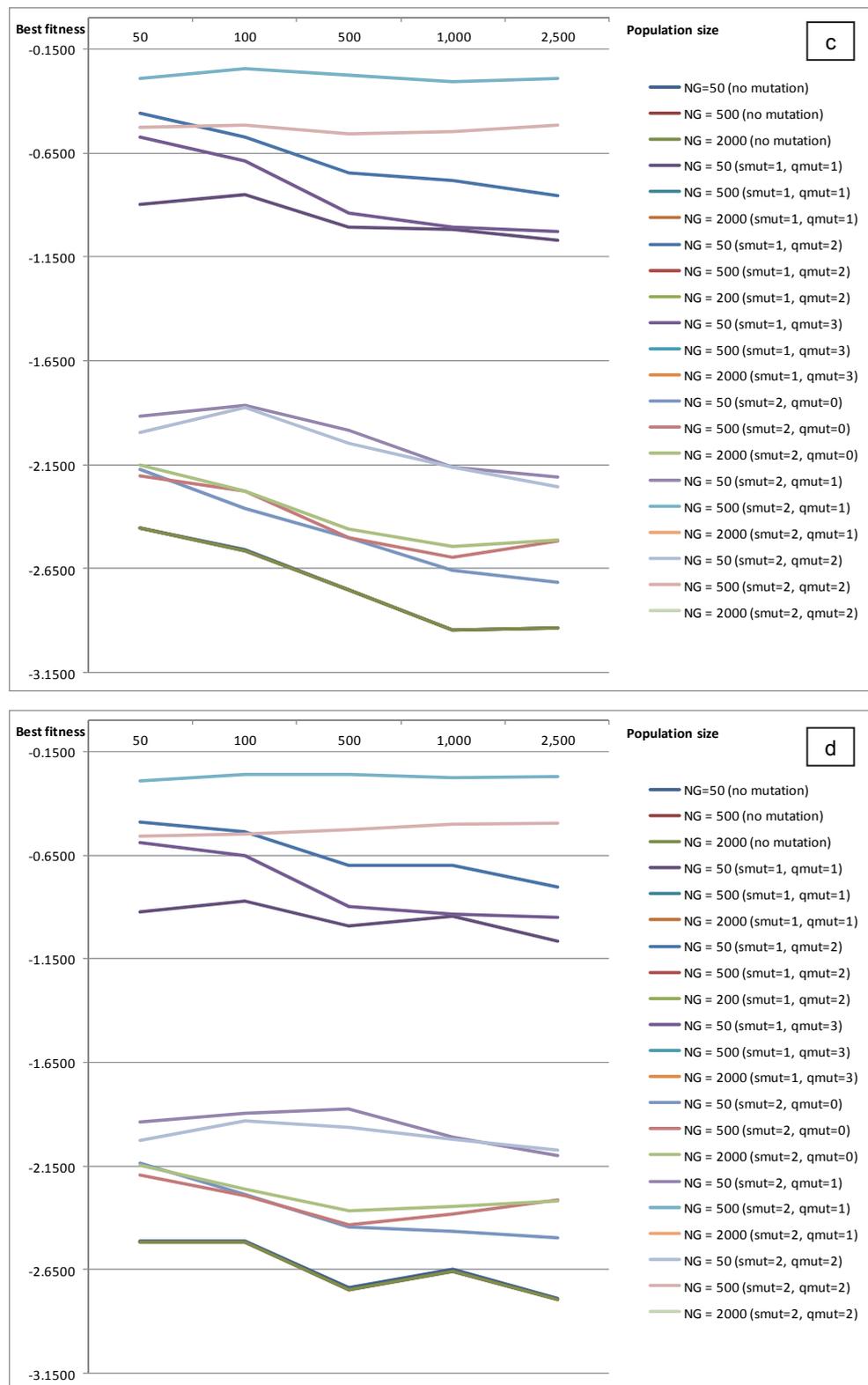


Figura 70. (continuación).

Las conclusiones que se derivan de la observación de estas figuras indican que:

- La política  $smut=1$  (desplazamiento en el cromosoma) arroja los peores resultados, como puede observarse en las gráficas (c) y (d) de la Figura 72, donde la dimensión del eje de ordenadas se ha tenido que ampliar para mostrar valores inferiores al resto de gráficas.

Incluso para los casos de las gráficas (a) y (d), los resultados son peores que los de las mismas gráficas de la Figura 73, que corresponden a la política  $smut=2$  (intercambio en el cromosoma).

- Los mejores resultados se obtienen en todos los casos para tamaños de población de 2.500 individuos ( $popsiz=2.500$ ). No obstante se aprecia que con 1.000 individuos se obtienen también resultados análogos, e incluso mejores en algunos casos. Para tamaños inferiores de la población a éstos, los resultados empeoran siempre.

Por tanto, y entendiéndose que cuantos más individuos tenga la población mayor es el coste computacional de generar la población inicial, se considera como el mejor valor para el tamaño de población el de  $popsiz=1.000$ .

- Con 50 generaciones se obtienen ya resultados buenos, que no mejoran sustancialmente al aumentar el número de generaciones (hay un claro estancamiento en la obtención del óptimo).

Por tanto, y siguiendo siempre la filosofía de ahorrar coste computacional para el algoritmo, se fija como valor idóneo el de  $NG=50$ .

- La política  $smut=2$  (intercambio de posición de genes en el cromosoma) obtiene mejores resultados siempre respecto a la otra política de mutación de desplazamiento, o en ausencia de mutación, para cualquier valor de  $qmut$ , aunque apreciándose una ligera mejoría con el valor de  $qmut=2$ .

Por tanto, se consideran como valores óptimos para el algoritmo genético los siguientes:  $smut=2$  y  $qmut=2$ .

- La política  $smut=2$  (intercambio de posición de genes en el cromosoma) obtiene mejores resultados siempre respecto a la otra política de mutación de desplazamiento, o en ausencia de mutación, para cualquier valor de  $qmut$ , aunque apreciándose una ligera mejoría con el valor de  $qmut=2$ .

Por tanto, se consideran como valores óptimos para el algoritmo genético los siguientes:  $smut=2$  y  $qmut=2$ .

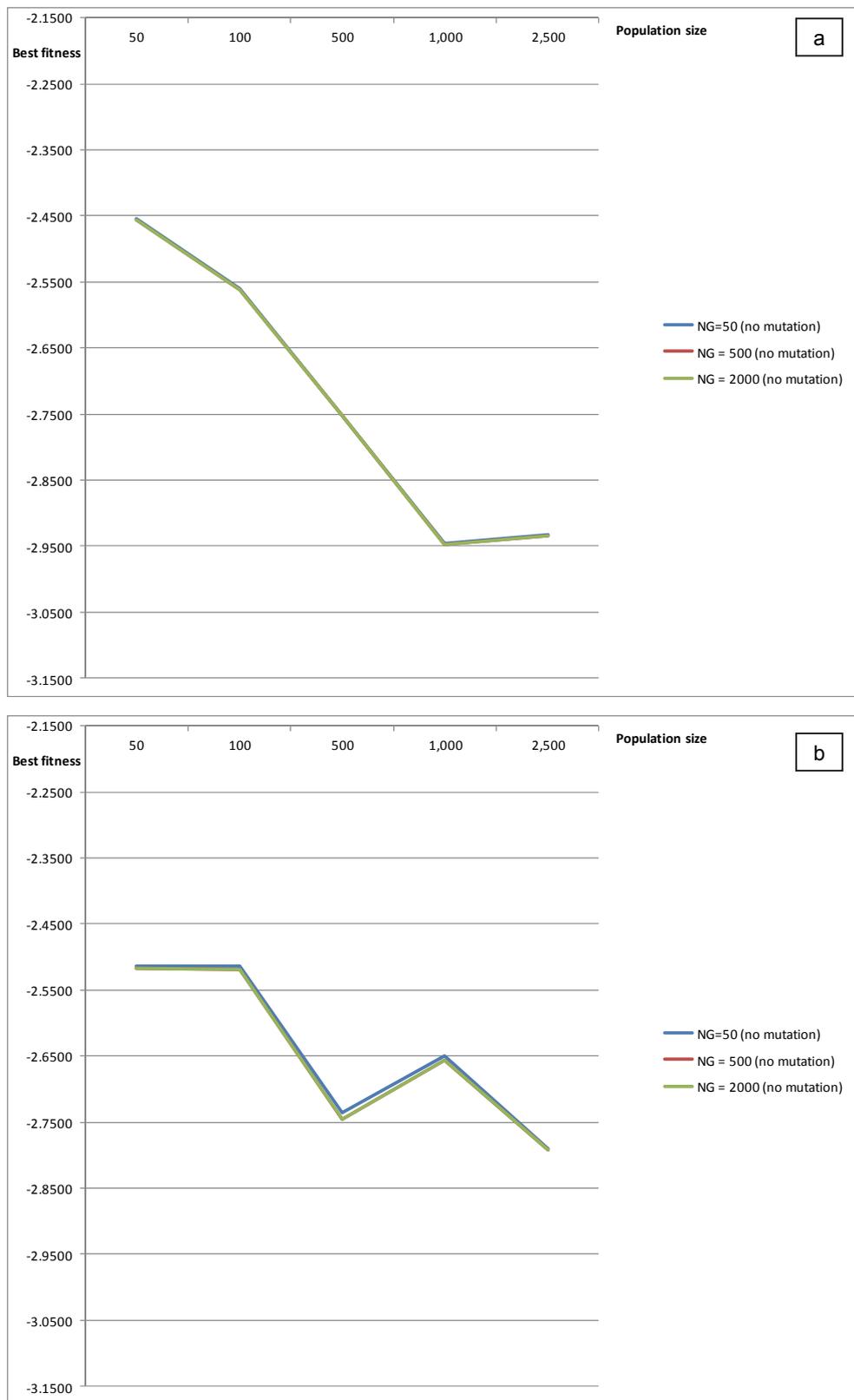


Figura 71. Extracción de resultados correspondientes a la Figura 70, mostrando solamente los casos que no consideran mutación.

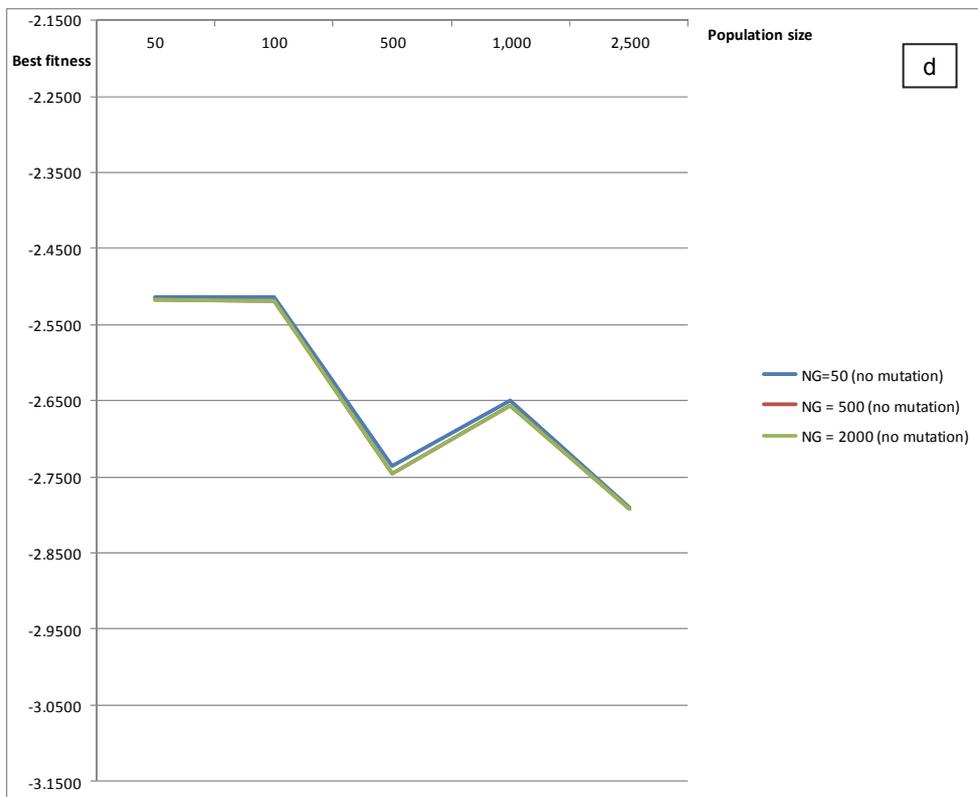
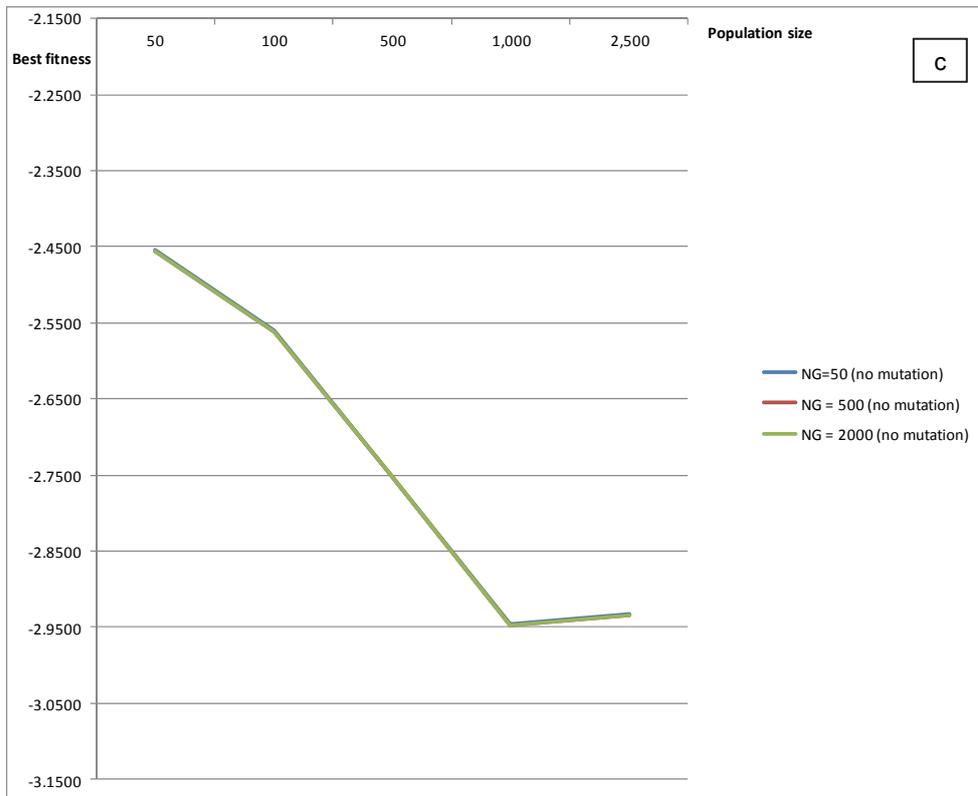


Figura 71. (continuación).

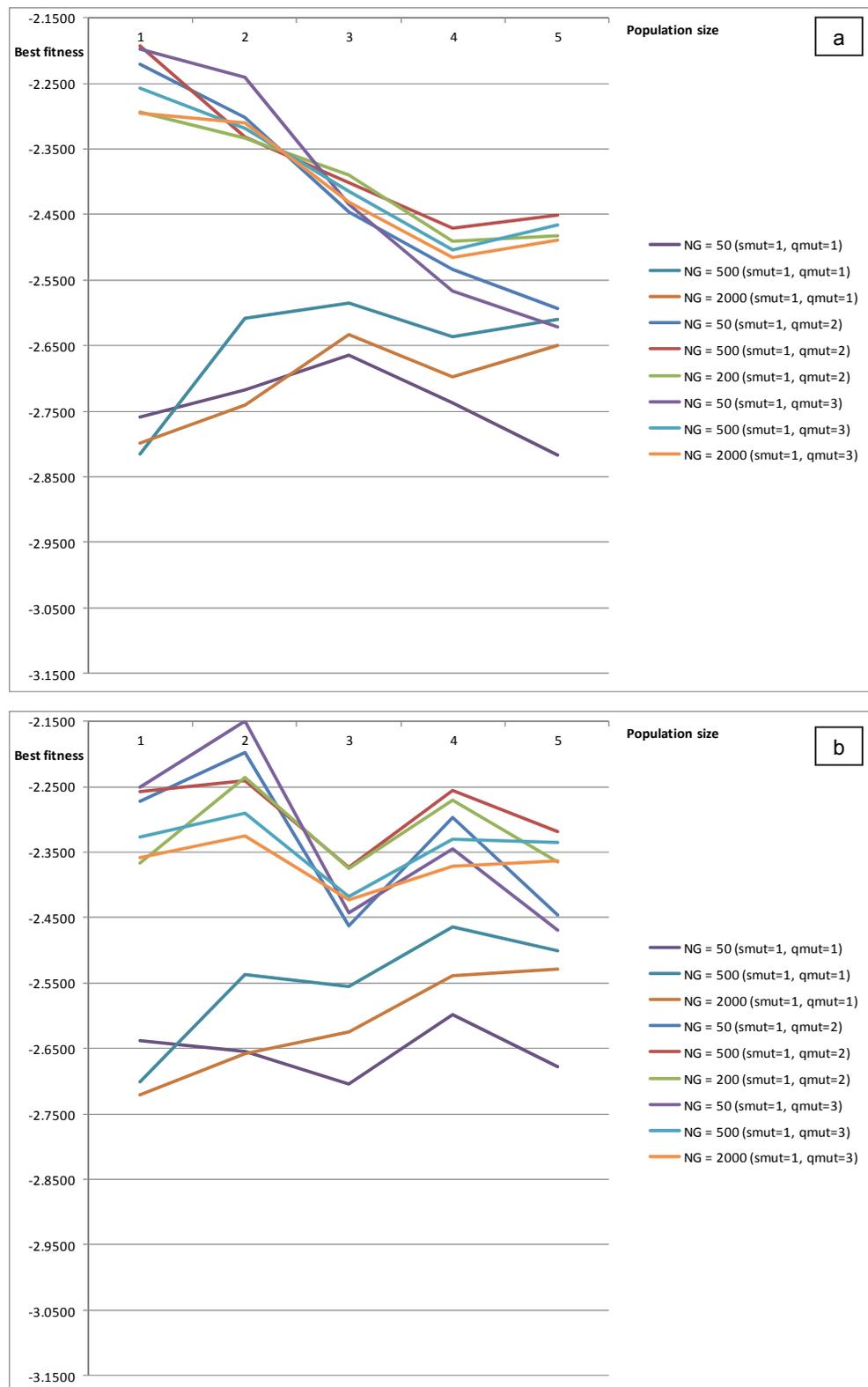


Figura 72. Extracción de resultados de la Figura 70, mostrando solamente los casos que consideran mutación  $smut=1$ . Importante: en (c) y (d), la dimensión del eje de ordenadas se ha ampliado para mostrar valores inferiores al resto de gráficas, denotando los malos valores obtenidos del *fitness*.

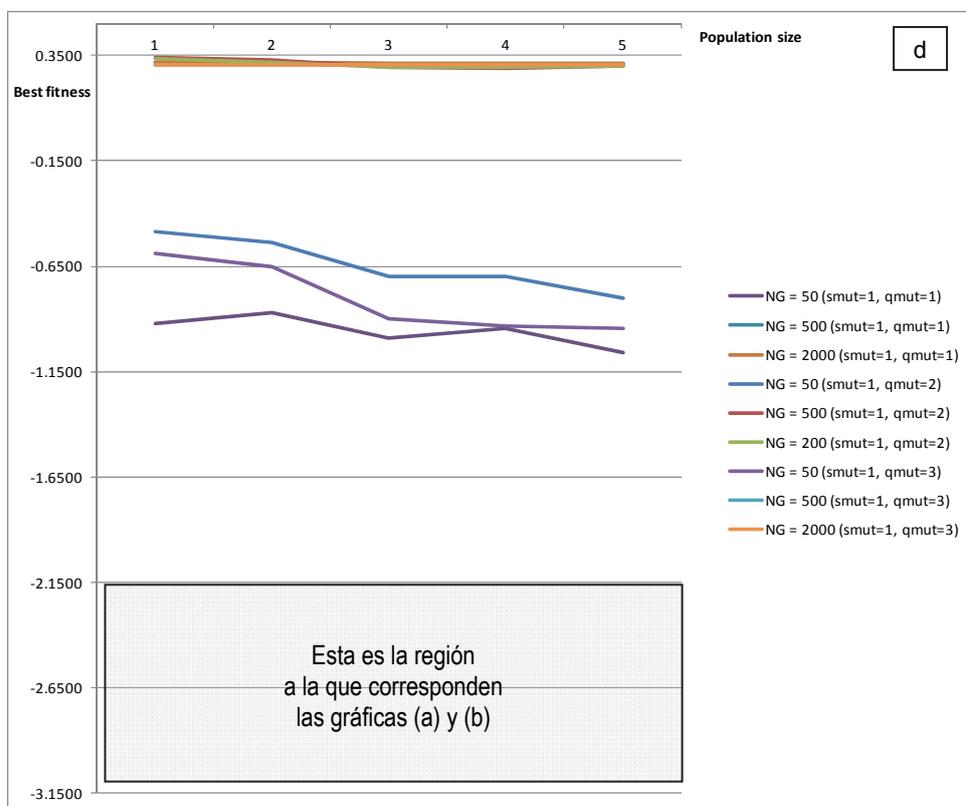
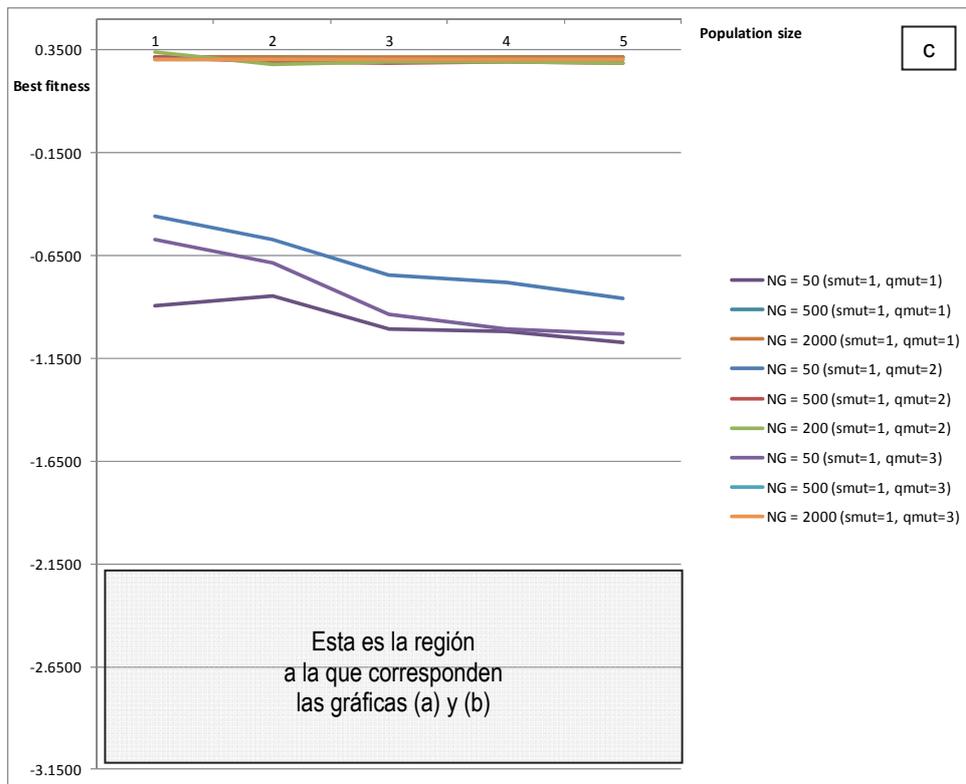


Figura 72. (continuación).

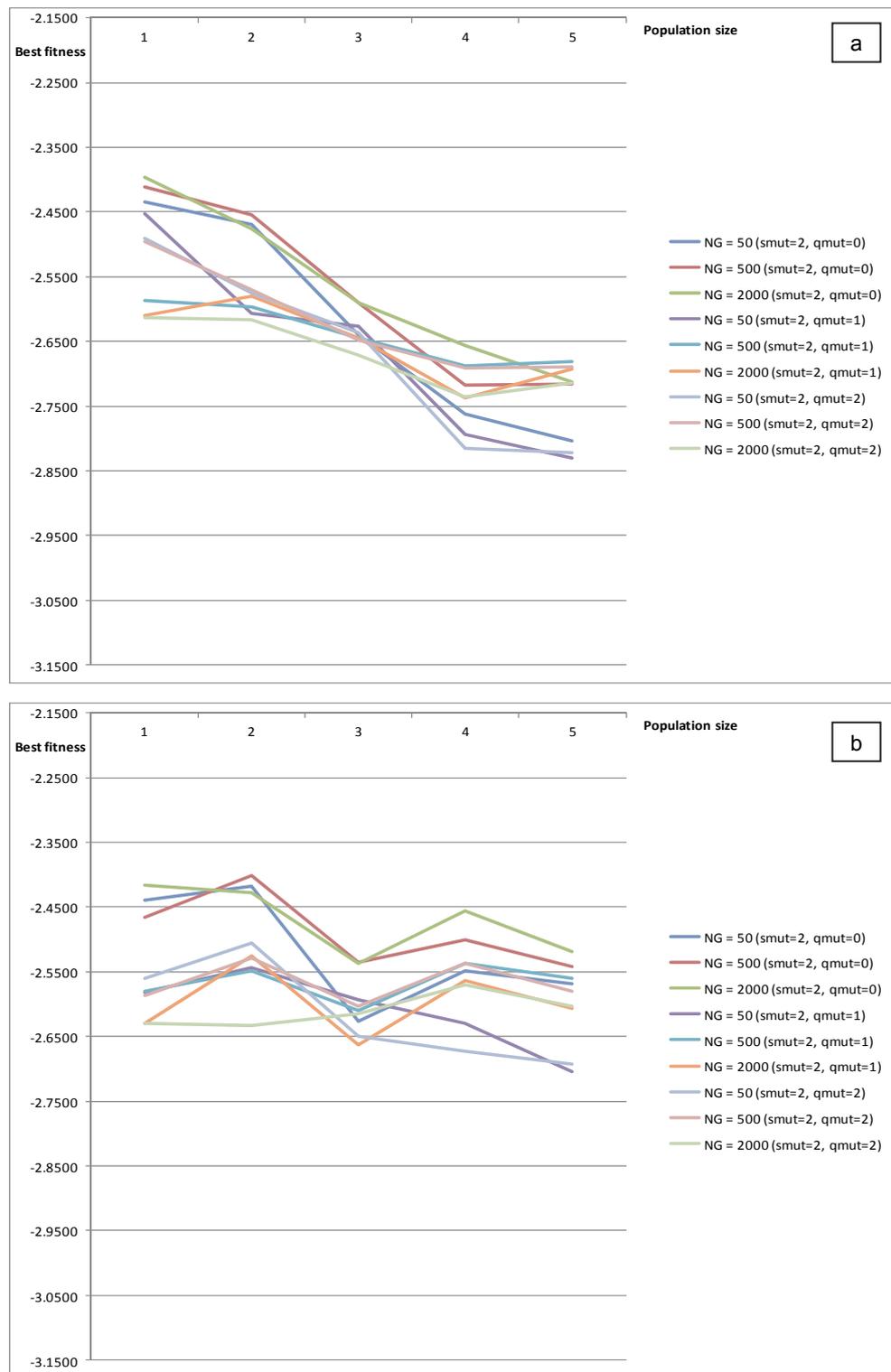


Figura 73. Extracción de resultados correspondientes a la Figura 70, mostrando solamente los casos que consideran mutación  $smut=2$ .

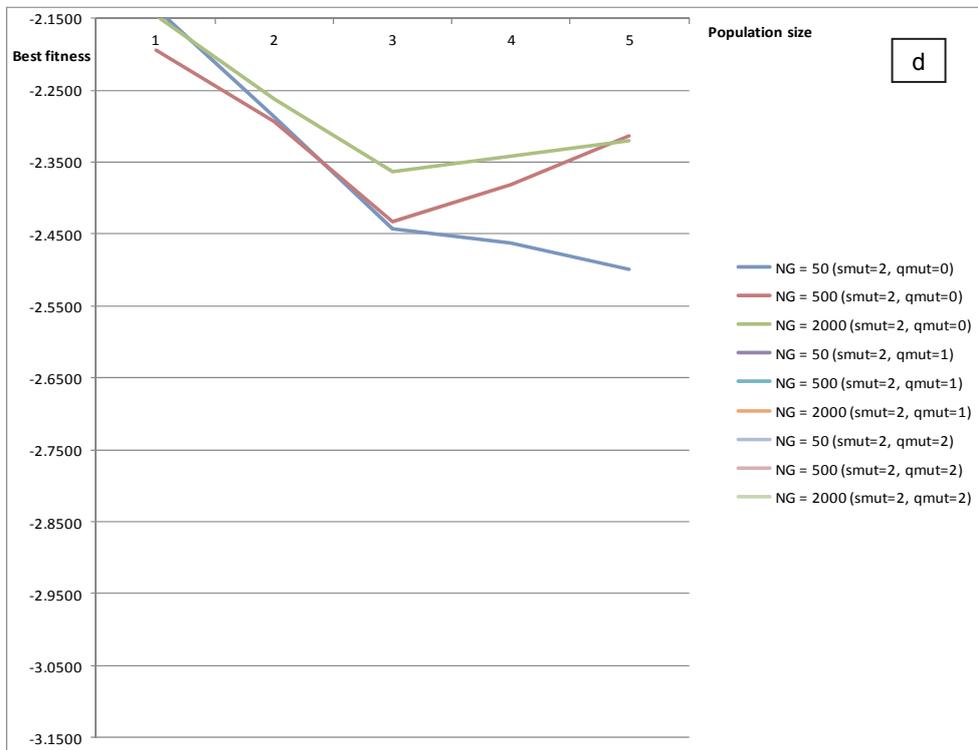
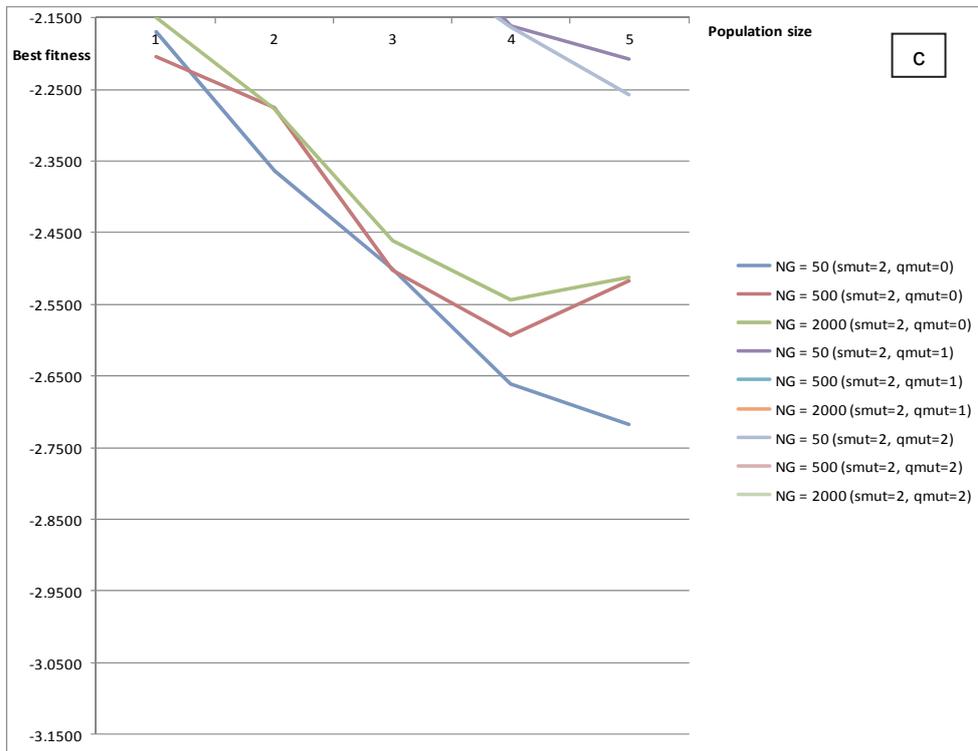


Figura 73. (continuación).

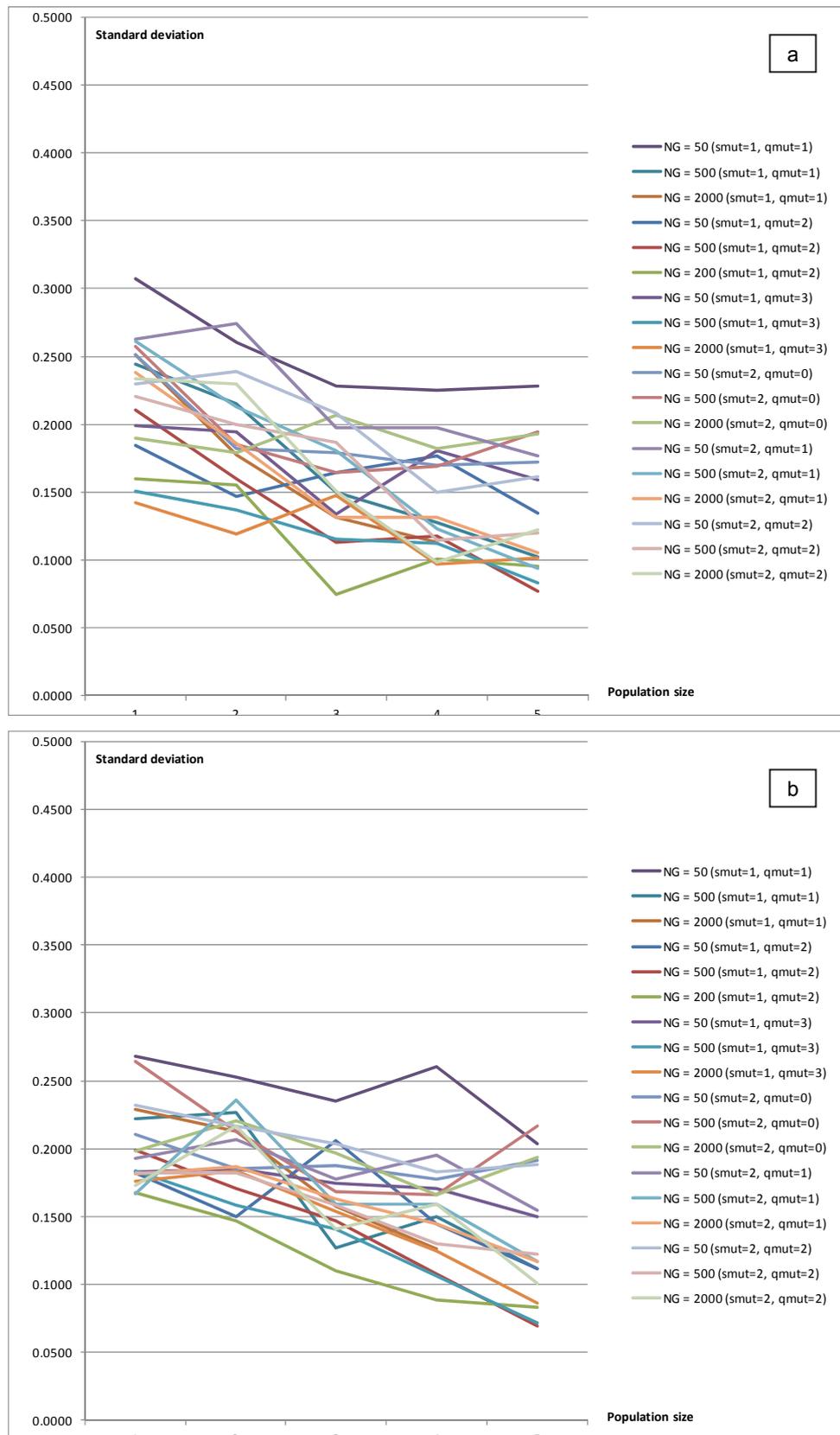


Figura 74. Desviación estándar de las soluciones óptimas encontradas, para los cuatro casos contemplados en la Figura 70.

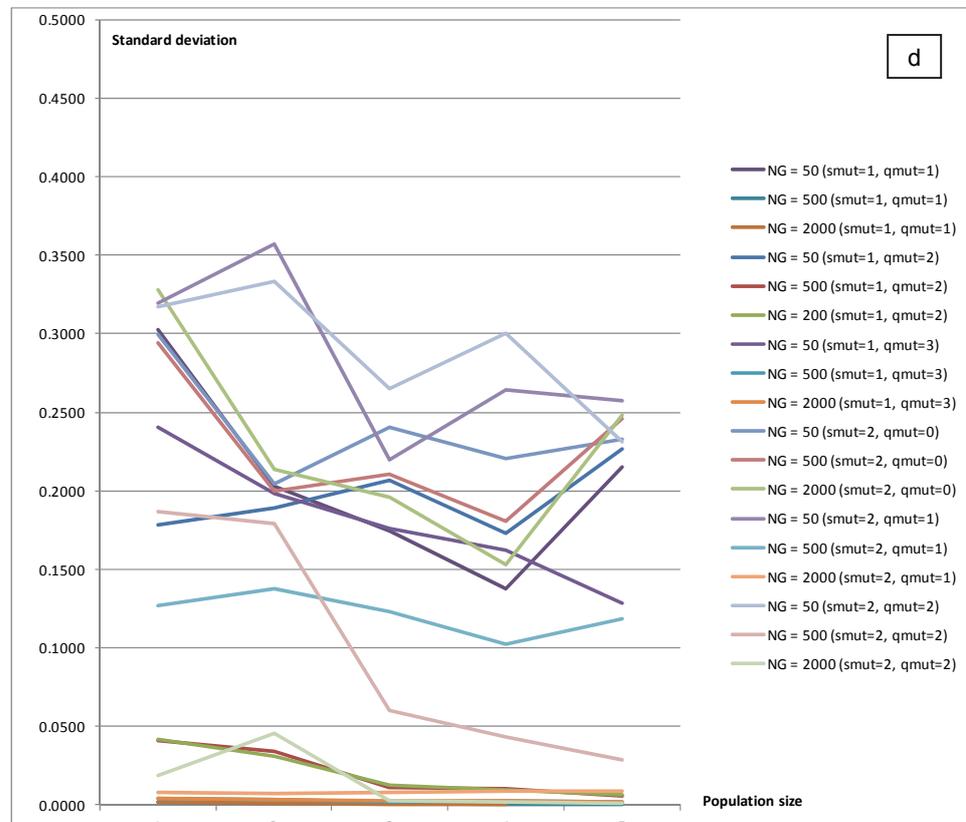
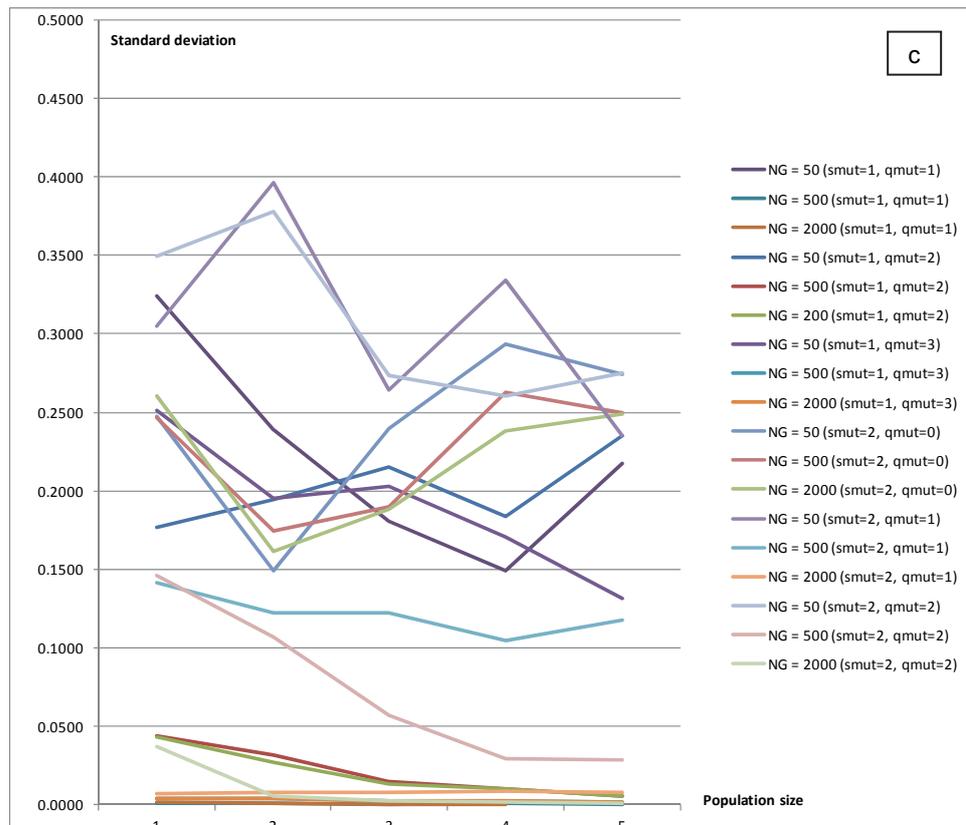


Figura 74. (continuación).

En resumen, los valores seleccionados para el ajuste paramétrico de GAVH son los que se muestran en la Tabla 21:

parámetro	valor
<i>NG</i>	50
<i>popsiz</i>	1.000
<i>h</i>	0,005
<i>pmut</i>	5
<i>smut</i>	2
<i>qmut</i>	2

Tabla 21. Valores ajustados de los parámetros del algoritmo genético GAVH, para la instancia experimental #1 considerando el perfil de usuario más genérico (*profile=1*), cuatro redes inalámbricas ( $NN=4$ ) y cuatro parámetros de calidad de servicio ( $NQ=4$ ).

Adicionalmente, se han realizado otros experimentos con para  $NQ=5$ , validándose las mismas conclusiones (pero con un coste computacional más elevado), por lo que el número de ejecuciones que se han efectuado en este estudio es muy superior al anteriormente citado de 21.000.

### 6.9.3.2 Políticas y porcentaje de mutación.

Una vez efectuado el barrido paramétrico anterior, y establecidos los valores ajustados de los parámetros de GAVH según la Tabla 21, se han realizado experimentos adicionales centrados en la mutación con el objetivo principal de ver la influencia del porcentaje de mutación.

En la Figura 75 se muestran los resultados de estos experimentos. Para la instancia experimental #1, y considerando 3 y 4 parámetros de calidad, se han evaluado las dos políticas de mutación (*smut=1*: mutación por desplazamiento; *smut=2*: mutación por intercambio) con distintas posiciones de genes (*qmut*).

En la Figura 76 se muestran los resultados de los mismos experimentos, pero para la instancia experimental #2. En ambas figuras se han señalado mediante flechas rojas los porcentajes óptimos en los casos en los que la existencia de mutación es más beneficiosa que no tenerla, considerando el caso *smut=2* y *qmut=2*, que es el ajuste fijado en la Tabla 21.

Del análisis de esta figura se pueden extraer las siguientes conclusiones:

- En general, cuanto mayor es el porcentaje de mutación, peores son los resultados, sea cual sea la política de mutación, instancia experimental o número de parámetros de calidad de servicio.
- En algunos casos, no aplicar mutación ofrece mejores resultados. No obstante, en otros casos la mutación es beneficiosa, teniendo en cuenta bajos porcentajes de mutación. Una decisión equilibrada para una pauta general sería considerar mutación con bajo porcentaje.

- A la vista de las figuras, y teniendo en cuenta los resultados de los experimentos del barrido paramétrico (mucho más exhaustivos que éstos), se corrobora como adecuado un 5% para el porcentaje de mutación (fijado en la Tabla 21), pues en las gráficas oscila entre el 2% y el 13%.

Hay que tener en cuenta que en los escenarios dinámicos de redes móviles heterogéneas, la variabilidad de parámetros de calidad y redes hace que puedan darse muchos casos en los que la mutación proporcione mejores o peores resultados. De aquí la necesidad de tener un valor para el porcentaje de mutación que, en la mayoría de los casos, se ha mostrado adecuado.

### **6.9.3.3 Consideraciones estadísticas**

La Figura 74 muestra las desviaciones estándar de las soluciones óptimas encontradas según el escenario experimental del barrido paramétrico mostrado en la Figura 70, excluyendo los tres casos de ausencia de mutación (donde no es necesario realizar varias ejecuciones del mismo experimento, al carecer en ese caso de ningún grado de aleatoriedad).

La desviación estándar es una medida de la dispersión de las soluciones, indica cuánto se alejan los valores respecto al promedio de las 50 ejecuciones. Por tanto, es una medida útil para determinar la volatilidad de los resultados y, de esta forma, la efectividad del algoritmo. Observando la Figura 74, se puede constatar que, en general, no hay volatilidades excesivas (solo pocos casos superan 0,35, teniendo en cuenta que los valores del *fitness* se mueven alrededor de -2,6, lo cual indica un máximo del 15% de desviación).

Se puede apreciar más uniformidad en las gráficas (a) y (b), siendo ligeramente mejores las desviaciones en (a), lo cual valida las conclusiones apuntadas anteriormente: sintonizar los valores  $h = 0,005$  y  $pmut = 5$  (se valora la uniformidad de todos los casos porque así se asegura un mismo comportamiento de GAVH en cualquier condición experimental).

### **6.9.4 Comparativa de rendimiento GAVH con SEFISA**

En el apartado 6.8.4.2 se mostró una comparativa de rendimiento de SEFI frente al algoritmo AHP [Lassoued 2008], al ser éste el único algoritmo sobre el que hay un trabajo publicado con instancias experimentales bien definidas y, por tanto, susceptible de ser comparado de forma realista.

De esta forma, la Tabla 20 mostraba los resultados obtenidos por los autores de dicho algoritmo AHP y los resultados que se han obtenido en esta tesis a partir de SEFI, mostrando las combinaciones de pesos que satisfacen la restricción (soluciones), para la instancia experimental #2 (6 redes inalámbricas y 5 parámetros de calidad de servicio).

Se ha aplicado el algoritmo GAVH para esta instancia experimental, considerando los mismos dos perfiles de usuario que en la comparación SEFI-AHP: perfil #2 (conversación) y perfil #3 (streaming).

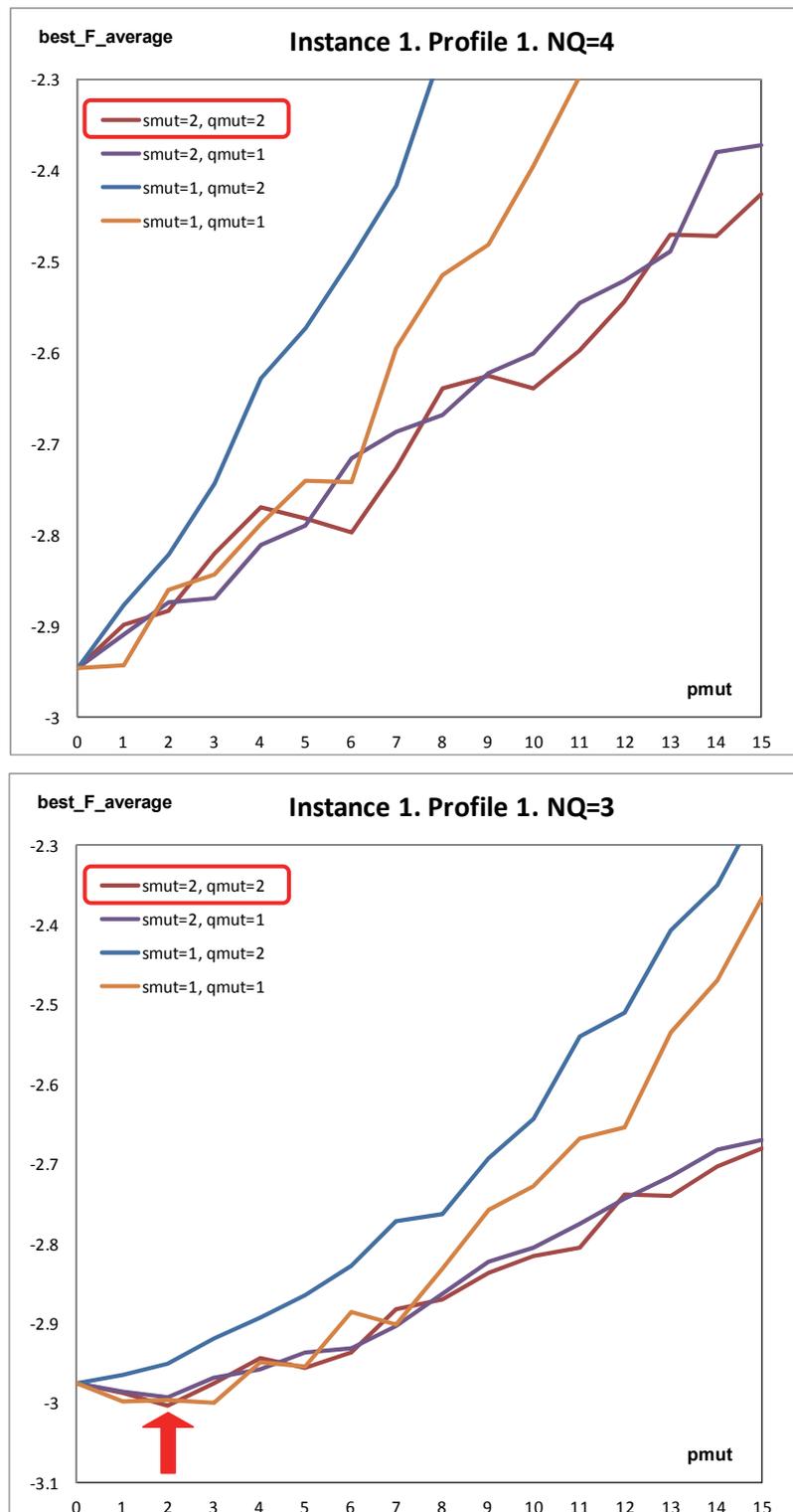


Figura 75. Influencia del porcentaje de mutación ( $pmut$ ). Para la instancia #1 y para 4 y 3 parámetros QoS, se han evaluado las dos políticas de mutación ( $smut$ ) con distintas posiciones de genes ( $qmut$ ). Las flechas rojas indican los porcentajes óptimos cuando es mejor tener una política de mutación a no tenerla, para el ajuste  $\{smut=2; qmut=2\}$ , que es el seleccionado según la Tabla 21, y que son cercanos al valor 5%, decidido en esa misma tabla.

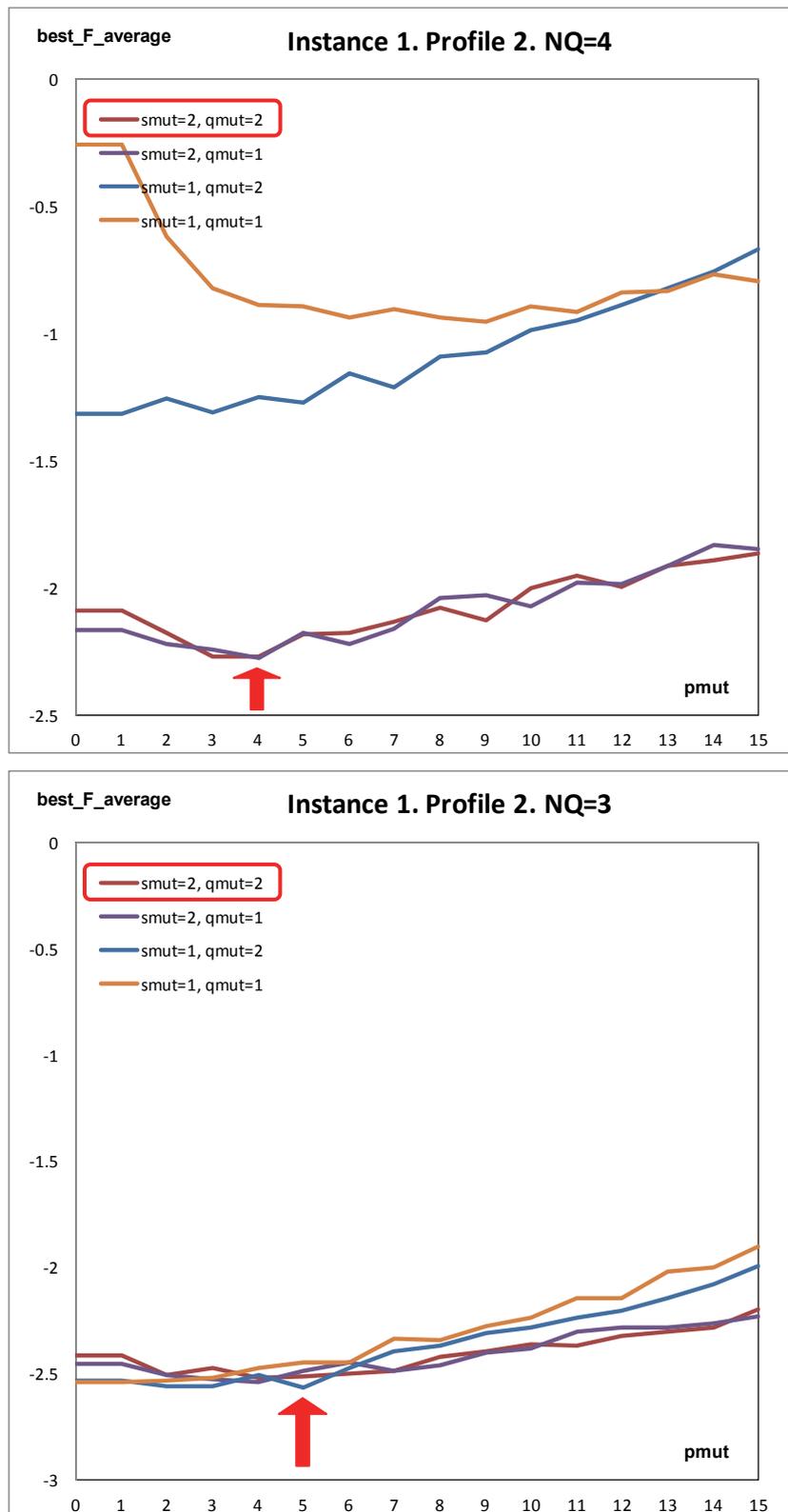


Figura 75. (continuación).

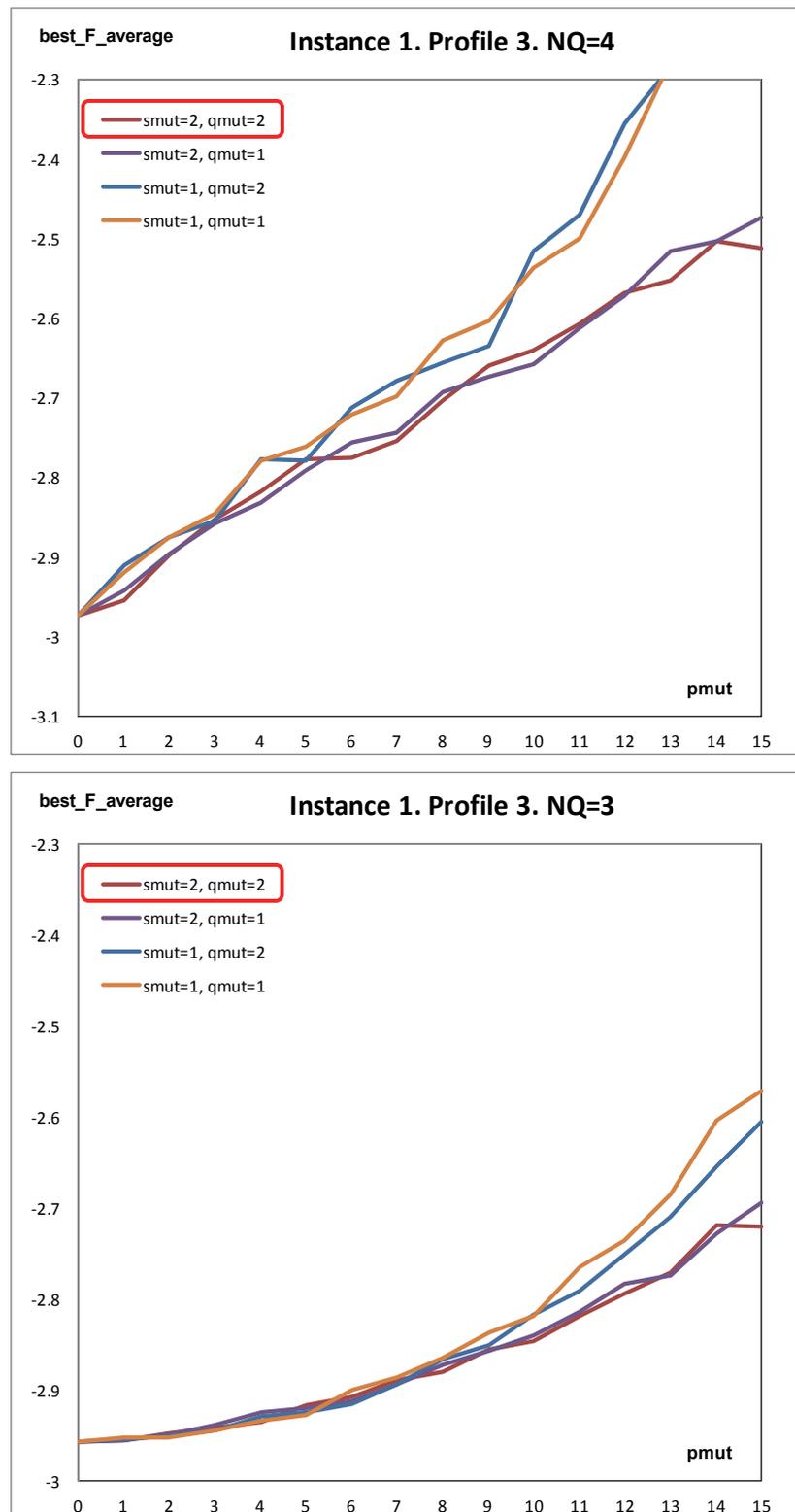


Figura 75. (continuación).

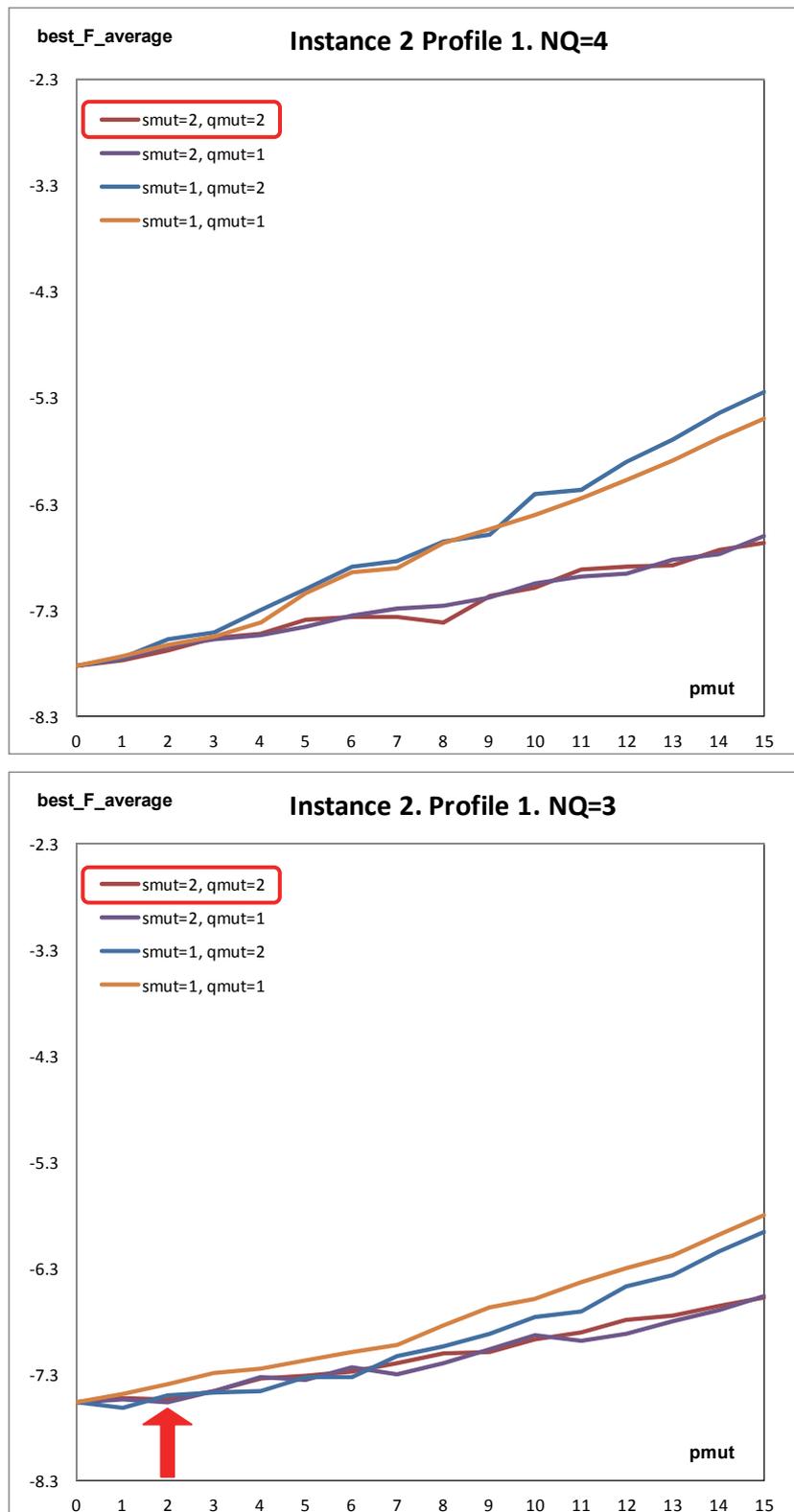


Figura 76. Resultados de los mismos experimentos que los de la Figura 75, pero en este caso para la instancia #2.

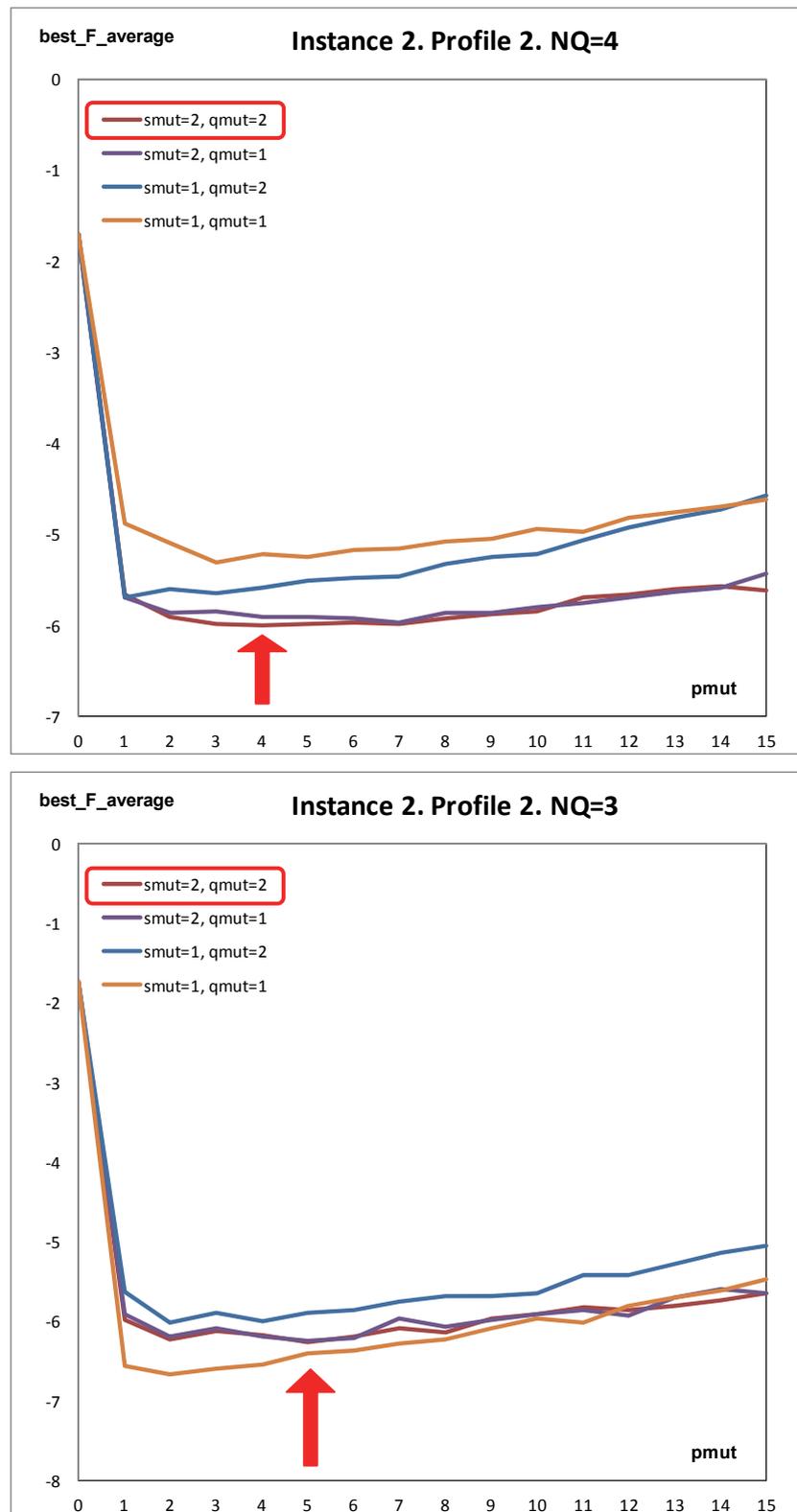


Figura 76. (continuación).

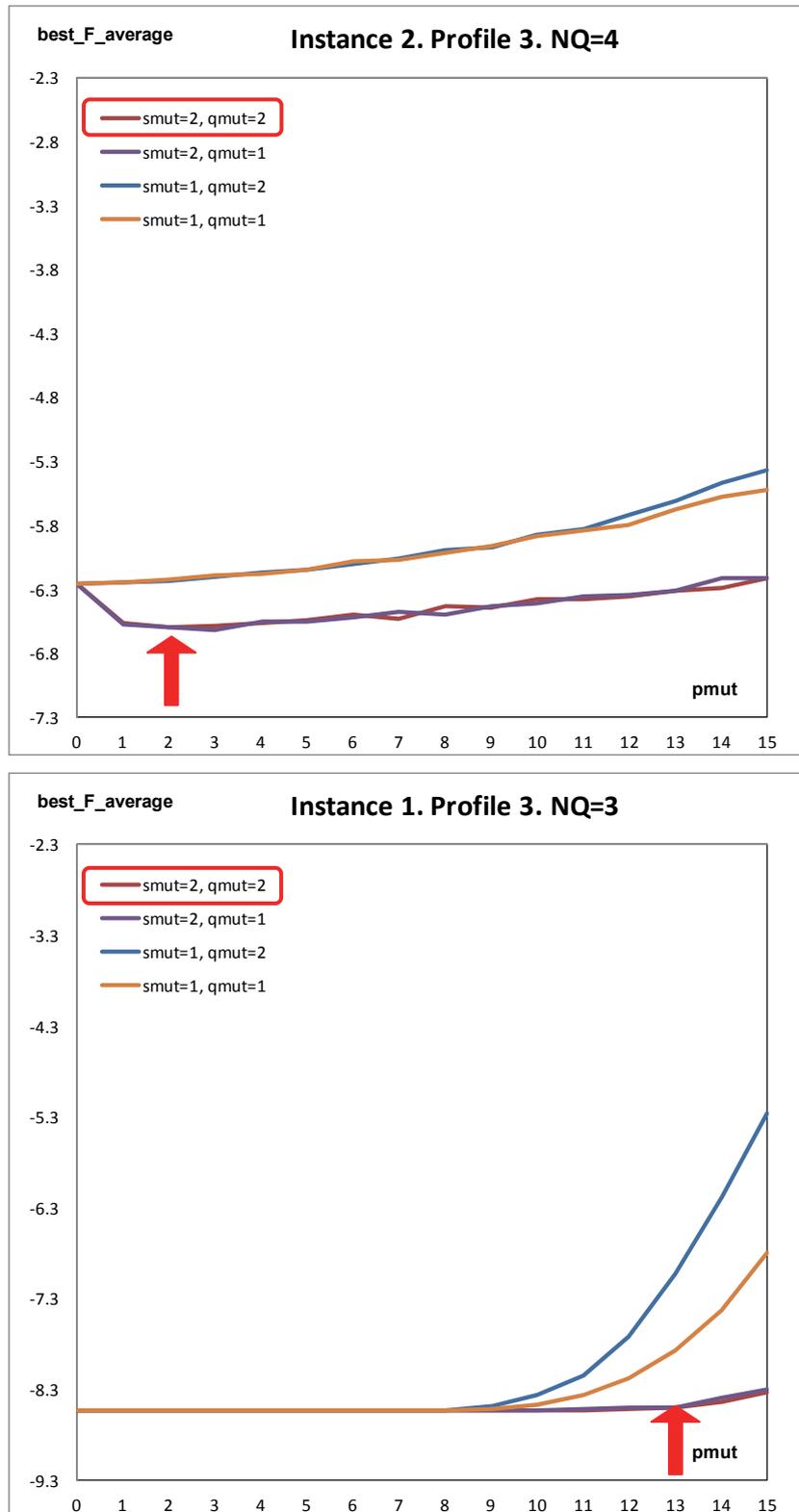


Figura 76. (continuación).

Los resultados obtenidos por GAVH se muestran en la Tabla 22, que es la Tabla 20 con las últimas líneas añadidas con sus resultados.

Es importante resaltar que los resultados del *fitness* de esta tabla para los experimentos con GAVH son el valor medio de las 50 ejecuciones del mismo experimento, mientras que la solución (vector de pesos) mostrada en la tabla corresponde con el *fitness* mínimo encontrado en esas 50 ejecuciones (no se puede mostrar el vector de pesos correspondiente al *fitness* promedio, ya que éste ha sido calculado de forma aritmética).

	Solución (pesos)					Red óptima	Fitness
	w0	w1	w2	w3	w4		
	B	E	D	S	C		
[Lassoued 2008] (perfil #2)	0.065	0.065	0.614	0.128	0.128	4	1.2596313
[Lassoued 2008] (perfil #3)	0.545	0.035	0.178	0.121	0.121	4	-4.428212
SEFI (h=0.01, perfil #2)	0.47	0.01	0.5	0.01	0.01	4	-2.08185
SEFI (h=0.005, perfil #2)	0.485	0.005	0.5	0.005	0.005	4	-2.17334
SEFI (h=0.01, perfil #3)	0.6	0.28	0.1	0.01	0.01	4	-6.03799
SEFI (h=0.005, perfil #3)	0.6	0.29	0.1	0.005	0.005	4	-6.0705
GAVH (h=0.01, perfil #2)	0.54	0.36	0.02	0.07	0.01	4	-5.7688
GAVH (h=0.005, perfil #2)	0.68	0.23	0.015	0.07	0.005	4	-5.8270
GAVH (h=0.01, perfil #3)	0.69	0.13	0.01	0.06	0.11	4	-6.4761
GAVH (h=0.005, perfil #3)	0.685	0.19	0.005	0.02	0.1	4	-6.5537

Tabla 22. Combinaciones de pesos que satisfacen la restricción (soluciones), para la instancia experimental #2 (6 redes, 5 parámetros QoS). Esta tabla completa la Tabla 20 añadiendo los resultados de GAVH. Las soluciones en [Lassoued 2008] han sido calculadas mediante el algoritmo AHP, para el perfil #2 (conversación, donde importa el retardo D:  $0.5 < D < 0.7$ ) y para el perfil #3 (streaming, donde importa un gran ancho de banda B y un retardo aceptable D:  $0.5 < B < 0.7$ ,  $0.1 < D < 0.3$ ). Las soluciones encontradas por GAVH mejoran sustancialmente los de SEFI, que ya eran buenos respecto a AHP. Por otro lado, el algoritmo GAVH proporciona mejores valores de *fitness* en todos los casos, aún con baja precisión de búsqueda.

Cabe destacar que, en el marco experimental de la comparativa GAVH-SEFISA-AHP, los tiempos de computación reportados por GAVH para una ejecución (50 generaciones y un tamaño de población de 1.000 individuos) han sido aproximadamente 0,8 segundos, lo cual satisface plenamente la restricción impuesta de tiempos de computación bajos, para permitir la aplicabilidad del algoritmo a dispositivos móviles.

### 6.9.5 Rendimiento de GAVH en procesadores empotrados

A la vista de los excelentes resultados ofrecidos por GAVH, este algoritmo sería nuestra propuesta para controlar la fase de decisión VH en dispositivos móviles. Dado que este tipo de dispositivos (smartphones, tablets, etc.) tienen limitadas sus capacidades de cómputo (microprocesadores con 1 GHz o menos) y es clave que la solución al problema sea obtenida minimizando tiempos de computación y consumos energéticos, se ha probado el algoritmo sobre un caso real de microprocesador empotrado de bajas prestaciones y reducido consumo eléctrico. Esta prueba es necesaria para verificar si es posible encontrar soluciones óptimas cercanas al tiempo real y con para este tipo de procesadores de bajo consumo.

La prueba ha sido llevada a cabo por medio del procesador MicroBlaze, un procesador “blando” (configurable a medida) implementado en FPGAs, muy utilizado como microprocesador empotrado en dispositivos electrónicos, y que posee unos rendimientos cercanos a los de muchos dispositivos móviles, como pueden ser los *smartphones* de baja gama (apenas un par de centenares de MHz y un par de vatios de potencia).

El procesador fue configurado para una frecuencia del reloj del sistema de 125MHz, unidad en coma flotante, sistema operativo *standalone* (autónomo), y una memoria local de 256KB que almacena el algoritmo. Fue implementado sobre una tarjeta de prototipado Digilent Virtex-5 XUPV5-LX110T (Figura 53) que lleva una FPGA Xilinx Virtex 5 XC5VLX110T-FF1136.

El rendimiento limitado de este procesador obligó a ajustar algunos de los parámetros de GAVH que fueron sintonizados (ver Tabla 21), principalmente el tamaño de la población y el número de generaciones, con el propósito de mantener el tiempo de respuesta por debajo de 1 segundo.

Una posibilidad para ayudar a contener el tiempo de computación consiste en utilizar poblaciones iniciales almacenadas previamente en arrays de datos, lo cual evita una considerable pérdida de tiempo en generar dichas poblaciones.

Reducir el tamaño de la población y/o el número de generaciones podría implicar una cierta pérdida de precisión de la solución obtenida, pero incluso así el resultado continúa siendo mucho mejor que las soluciones reportadas por AHP y el resto de los algoritmos.

Por ejemplo, si se considera el caso mostrado en la Tabla 22 para  $DS2$ ,  $NN = 4$ ,  $NQoS = 5$ , perfil de usuario  $P2$ , y  $h = 0,01$ , donde el fitness para GAVH era  $-5.77$ , ahora obtenemos un fitness medio de  $-4.36$  al reducir tanto el tamaño de la población como el número de generaciones a tan solo 20 individuos y generaciones respectivamente. Nótese que, en este caso, el resultado sigue siendo mucho mejor no solo que AHP (1,26), sino también que los ofrecidos por los algoritmos SEFI (-2,08) y SEFISA (-1,89).

### 6.9.6 Comparativa de rendimiento con otros autores

En [Nkansah 2007] se propone el algoritmo **FMADM** (*fuzzy multiple attribute decision making*), que es un algoritmo inteligente de tipo MCDM (ver apartado 6.2.2). Este algoritmo utiliza la lógica borrosa para hacer una selección óptima de los pesos de los parámetros QoS en función del perfil de usuario requerido.

El algoritmo FMADM utiliza la instancia #3, claramente definida en [Nkansah 2007], y perfiles de usuario para dos casos de utilización, descarga de archivos y videoconferencia, donde los pesos correspondientes a ciertos parámetros adquieren una mayor o menor importancia.

En esta tesis se ha definido el perfil #4 (con dos casos o sub-perfiles) y el perfil #5 para que correspondan con los casos reflejados en [Nkansah 2007], ya que en dicho artículo no se explicitan los rangos de pesos de los perfiles, pues la técnica de lógica borrosa que utiliza se encarga tan solo de respetar una precedencia de importancia en los parámetros, en lugar de establecer rangos. Por ejemplo, en este artículo se establece que en el perfil #4 el parámetro más importante es la tasa de transferencia de datos (ancho de banda), seguido por el coste económico, y en el perfil #5 el parámetro más importante es la tasa de transferencia de datos (ancho de banda), seguido por el coste económico y la latencia de red, por este orden.

Se han efectuado diversos experimentos con los algoritmos SEFI y GAVH para comparar sus resultados con los ofrecidos por la heurística FMADM, y de esta forma seguir comprobando o validando el rendimiento que proporcionan las metodologías desarrolladas en esta tesis.

En estos experimentos se han utilizado los parámetros de GAVH sintonizados según la Tabla 21, pero con un valor de  $h=0,02$ , ya que el muy elevado número de parámetros QoS que tiene la instancia #3 (10 parámetros) eleva los tiempos de computación por encima de los requisitos de tiempo real necesarios para la utilización en dispositivos móviles si se utilizan valores muy bajos de  $h$ . Con este valor de  $h=0,02$  (también utilizado para SEFI) se consiguieron tiempos cercanos a las dos décimas de segundo, tanto en SEFI como en GAVH.

Las soluciones óptimas encontradas por FMADM para los dos perfiles de usuario se muestran en la Tabla 23, y son tomadas de [Nkansah 2007]. Asimismo, esta tabla muestra las soluciones óptimas encontradas por los algoritmos SEFI y GAVH, correspondientes a la misma red WLAN\_P\_2 que la considerada por FMADM. En esta tabla se pueden observar los valores correspondientes a los pesos de los 10 parámetros QoS, explicados en el apartado 6.4.

Los valores de *fitness* correspondientes a estas soluciones óptimas se muestran en la Tabla 24. En esta tabla se muestran dos valores de *fitness*, correspondientes a sus dos formulaciones:

- **Fitness F**. Este *fitness* corresponde a la formulación dada en (29), que es la considerada en esta tesis, y que se ha utilizado a lo largo de este capítulo en los distintos experimentos y comparativas de SEFI, SEFISA, GAVH y AHP. El objetivo de estos algoritmos de optimización es minimizar el valor de F.

- Fitness **WNSF**. Este *fitness* corresponde con la formulación según el trabajo realizado en [Nkansah 2007]. Como se explicó en el apartado 6.4.3, esta función de *fitness* es maximizada por sus autores, al contrario que F. Si bien las formulaciones de F y WNSF son las mismas, difieren en el criterio dado en la ecuación (34), por lo que **WNSF = -F**.

Peso:	w0	w1	w2	w3	w4	w5	w6	w7	w8
Parámetro QoS:	S	D	A	L	R	E	P	V	C
FMADM perfil=4	0,0475	0,4587	0,0475	0,0475	0,0475	0,0475	0,0475	0,0475	0,209
FMADM perfil=5	0,0402	0,4528	0,0402	0,0981	0,0402	0,0402	0,0402	0,0402	0,2081
GAVH perfil=4.A	0,06	0,64	0,02	0,02	0,02	0,02	0,02	0,02	0,18
GAVH perfil=4.B	0,02	0,5	0,02	0,02	0,02	0,02	0,02	0,02	0,36
GAVH perfil=5	0,14	0,46	0,02	0,02	0,02	0,02	0,02	0,02	0,28
SEFI perfil=4.A	0,02	0,66	0,02	0,02	0,02	0,02	0,02	0,02	0,2
SEFI perfil=4.B	0,02	0,56	0,02	0,02	0,02	0,02	0,02	0,02	0,3
SEFI perfil=5	0,02	0,48	0,02	0,1	0,02	0,02	0,02	0,02	0,3

Tabla 23. Soluciones (combinaciones de pesos que satisfacen la restricción) generadas por los algoritmos FMADM, GAVH y SEFI, utilizando la misma instancia experimental y perfiles de usuario.

Del análisis de los resultados mostrados en la Tabla 24, se puede concluir:

- **SEFI y GAVH mejoran siempre (perfiles #4 y #5) los resultados de FMADM.**
- **GAVH mejora (perfil #5) o iguala (perfil #4) los resultados de SEFI.**

### 6.9.7 Conclusiones

Del análisis de la Tabla 22 cabe destacar como resultado principal que **las soluciones encontradas con GAVH tienen mucho mejor *fitness* que SEFI en todos los casos, y por tanto mejora a SEFISA y AHP sustancialmente, aún con una baja precisión de búsqueda.**

Por tanto, **GAVH emerge como la mejor opción para la fase de decisión VH**, de todas las opciones encontradas hasta el momento.

El algoritmo GAVH ha sido sintonizado en sus parámetros (destacando el hecho de considerar bajos porcentajes de mutación) y evaluado su rendimiento bajo distintas condiciones experimentales, repitiéndose en todos los casos un excelente comportamiento del algoritmo.

Por otro lado, los resultados de la comparación GAVH-SEFI-FMADM que se muestran en la Tabla 24 no hacen sino confirmar los ya obtenidos en la comparativa de GAVH con SEFISA y AHP.

En síntesis, **todas las comparativas efectuadas apuntan a GAVH como el algoritmo idóneo para resolver en tiempo real el problema de optimización consistente en la selección basada en QoS de la mejor red inalámbrica para establecer una nueva conexión en dispositivos móviles.**

	Función WNSF			Función F		Combinaciones	Soluciones	Varianza	Desviación estándar
	UMTS_1	WLAN_P_1	WLAN_P_2	F óptimo	Red opt.				
FMADM perfil=4	0,4373	0,6973	0,7843						
FMADM perfil=5	0,4300	0,7098	0,7992						
GAVH perfil=4.A	0,3200	0,7252	0,8512	-0,8512	2	24.255	12.825	7,07E-07	8,41E-04
GAVH perfil =4.B	0,3780	0,7532	0,8512	-0,8512	2	24.310	12.870	7,04E-07	8,39E-04
GAVH perfil =5	0,4340	0,7612	0,8512	-0,8512	2	48.180	23.980	3,62E-06	1,90E-03
SEFI perfil =4.A	0,2980	0,7212	0,8512	-0,8512	2	24.255	12.825		
SEFI perfil =4.B	0,3480	0,7412	0,8512	-0,8512	2	24.310	12.870		
SEFI perfil =5	0,3880	0,7492	0,8472	-0,8472	2	48.180	23.980		

GAVH y SEFI mejoran el resultado de FMADM en todos los casos

GAVH mejora el resultado de SEFI para el perfil #5

GAVH y SEFI dan resultados similares en el caso del perfil #4

Tabla 24. Valores de las funciones de fitness según las formulaciones WNSF y F, correspondientes a las soluciones generadas por los algoritmos FMADM, GAVH y SEFI que se muestran en la Tabla 23 .

## **6.10 Conclusiones finales**

En este trabajo de tesis se han presentado propuestas algorítmicas originales, inmersas en la llamada “inteligencia computacional”, que resuelven el problema de optimización de la fase de decisión del *Vertical Handoff* en la conectividad de terminales móviles en un escenario de redes inalámbricas heterogéneas.

Se han propuesto una heurística exacta no exhaustiva (SEFI), una adaptativa basada en *Simulated Annealing* (SEFISA) y una basada en un algoritmo genético (GAVH). La primera ha sido ideada principalmente para evaluar el coste del problema, realizar una primera aproximación a la búsqueda de soluciones (cubriendo de forma uniforme todo el espacio de soluciones) y generar las poblaciones de los otros dos algoritmos.

Los algoritmos SEFISA y GAVH perfeccionan el método de obtención de soluciones óptimas básicamente al satisfacer las restricciones impuestas por la aplicabilidad real de las técnicas software requeridas: programas que sean ligeros y muy rápidos en proporcionar soluciones óptimas, al tener que ser ejecutados en procesadores de bajo rendimiento y consumo en entornos de decisión rápida próximos al tiempo real.

Bajo estas condiciones, GAVH emerge como el algoritmo que mejor soluciones proporciona, bajo cualquiera de las instancias experimentales y perfiles de usuario que se han considerado, y tanto frente a SEFISA (algoritmo propio) como frente a AHP y FMADM (que son algoritmos propuestos por otros autores para abordar este mismo problema de optimización).

## 6.11 Aportaciones

Como aportaciones realizadas por la doctoranda en este problema de optimización de la fase de decisión *Vertical Handoff* con dispositivos móviles en redes inalámbricas heterogéneas, se citan las siguientes:

- Estudio y definición del problema de optimización, así como su adaptación para poder ser abordado mediante técnicas evolutivas, especialmente mediante la propuesta de codificación de las posibles soluciones.
- Desarrollo de una técnica de búsqueda directa (SEFI) como base de experimentación para la búsqueda de soluciones aproximadas y para la configuración de poblaciones en los algoritmos evolutivos desarrollados (SEFISA y GAVH).
- Propuesta del algoritmo GAVH como solución computacional para resolver el problema. Este algoritmo ha podido ser desarrollado gracias a la experiencia adquirida en el desarrollo de otros algoritmos genéticos en dos trabajos de esta tesis: el problema de optimización de la predicción de series temporales y el problema de optimización de la calidad de los códigos correctores de errores en comunicaciones.
- Sintonizar los algoritmos y estudiar su comportamiento mediante la realización de numerosos experimentos, y evaluar sus rendimientos al compararlos con otras técnicas heurísticas propuestas para resolver el problema. Identificar potencialidades de aplicación real y definir líneas de trabajo futuro.
- Adaptar los parámetros de algoritmo propuesto para contener los tiempos de ejecución en un microprocesador empujado basado en FPGA (Microblaze), que como procesador de bajo consumo y rendimiento permite hacer una aproximación a los disponibles en muchos dispositivos móviles.

Las anteriores aportaciones aparecen en las siguientes publicaciones:

- **Revistas.**
  - "Fast Decision Algorithms in Low-Power Embedded Processors for Quality-of-Service Based Connectivity of Mobile Sensors in Heterogeneous Wireless Sensor Networks". **María D. Jaraíz-Simón**, Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez and Juan M. Sánchez-Pérez. Sensors, Volume 12, Issue 2, MDPI AG, Basel, Switzerland, 2012, pag:1612-1624, ISSN:1424-8220. (Impact factor = 1.953 -Q1-).
  - "Embedded Intelligence for Fast QoS-based Vertical Handoff in Heterogeneous Wireless Access Networks". **Maria D. Jaraiz-Simon**, Juan A. Gomez-Pulido, Miguel A. Vega-Rodríguez. Pervasive and Mobile Computing, Elsevier Science, Amsterdam, The Netherlands, 2014, pp.:1-15, ISSN:1574-1192. (Impact factor = 1.629-12 -Q1-)

- **Capítulos de libro.**

- “Simulated Simulated Annealing for Real-Time Vertical-Handoff in Wireless Networks”, in: "Advances in Computational Intelligence. Lecture Notes in Computer Science, Vol. 7902". **María D. Jaraíz-Simón**, Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez, and Juan M. Sánchez-Pérez. I. Rojas, G. Joya, J. Gabestany (Eds.). Springer-Verlag, Berlin, Germany, 2013, pp.:198-209. ISBN:978-3-642-38678-7.

- **Conferencias.**

- “Algoritmo Genético Empotrado en Dispositivos Móviles para Control de Conectividad en Redes Inalámbricas Heterogéneas”. **María D. Jaraíz-Simón**, Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez. Actas del XV Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 13) - IX Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2013), , Madrid, España, 2013, pp.:683-692. ISBN:978-84-695-8348-7.
- "Procesador Dual Embebido para Vertical Handoff basado en Calidad del Servicio". Juan A. Gómez-Pulido, **María D. Jaraíz-Simón**, Miguel A. Vega-Rodríguez, Juan M. Sánchez-Pérez, Carlos T. Calafate. Actas del I Simposio en Computación Empotrada, SICE 2010, Ibergarceta Publicaciones S.L., Valencia, España, 2010, pag: 137-144. ISBN:978-84-92812-69-1.

# ***7 Principales aportaciones y trabajos futuros***

*“Soy de las que piensan que la ciencia tiene una gran belleza. Un científico en su laboratorio no es sólo un técnico: es también un niño colocado ante fenómenos naturales que le impresionan como un cuento de hadas.”*

**Marie Curie.**

Física, química y matemática polaca, nacionalizada francesa.  
(1867 - 1934).

## **7.1 Principales aportaciones**

A continuación se exponen de forma muy resumida las principales aportaciones derivadas de los trabajos de esta tesis doctoral y que, en líneas generales, aportan una visión positiva a la aplicación de la computación evolutiva para resolver problemas de optimización combinatoria en el ámbito de las comunicaciones móviles inalámbricas, que era el objetivo global de la tesis.

### **Problema TS.**

- Se ha definido un problema de optimización en cuanto a establecer el parámetro forgetting-factor como objetivo a optimizar, frente a otros posibles parámetros, principalmente la dimensión del modelo.
- Se realizaron numerosos experimentos con el objetivo de verificar, para distintos benchmarks de series temporales, la sensibilidad de la identificación frente a la variabilidad del tamaño del modelo y del forgetting-factor.
- Se buscaron y establecieron umbrales límites para el forgetting-factor, así como los valores de referencia para el tamaño de los modelos.
- Se propuso un algoritmo genético para resolver el problema de optimización, y se implementó una versión paralela para mejorar los resultados del anterior.
- Se obtuvo una efectiva mejoría, aunque mínima, de la propuesta de implementación del algoritmo genético paralelo frente al algoritmo genético clásico. Si bien la mejoría es mínima, en algunos casos la precisión requerida a los modelos puede ser esencial, sobre todo para series temporales de sistemas con tiempo de muestreo muy pequeño o con necesidades de predicción críticas por cuestiones de seguridad.

### **Problema ECC.**

- Se propuso, ante problemas con instancias que requieren un alto coste computacional, una metodología de experimentaciones previas con instancias más pequeñas (y por tanto, de computación más rápida), que permitió ajustar rápidamente los parámetros del AG. De esta forma, el análisis de los resultados en el problema de optimización de los códigos correctores de errores podría aplicarse a casos de instancias más grandes.
- Se implementó una propuesta de AG con sus principales parámetros bien sintonizados, lo cual permitió mejorar las marcas obtenidas por otras heurísticas evolutivas publicadas por otros autores.

### **Problema MANET.**

- Se utilizaron dos metaheurísticas para resolver este problema de optimización multi-objetivo que tiene cinco variables de decisión y tres objetivos: NSGA-II y EDM. Los experimentos realizados revelan que NSGA-II produce mejores resultados, según la métrica hipervolumen, aunque los valores de las métricas parecen indicar que ambos algoritmos producen frentes muy próximos.

- Se utilizó computación paralela para mejorar los frentes de Pareto del problema y optimizar el rendimiento de los algoritmos, calculando más evaluaciones de la función en un plazo razonable de tiempo.
- Se definió una propuesta de optimización paralela multi-objetivo para el problema, según el modelo de islas. Este enfoque paralelo permitió conseguir resultados óptimos minimizando el tiempo. Los resultados así logrados animan a aplicar esta técnica a otros escenarios MANET.
- Los resultados del hipervolumen obtenidos por EDM han sido razonablemente buenos cuando se comparan con los aportados por otras heurísticas de optimización multi-objetivo: si bien no es la mejor heurística, está en la línea de buenos resultados y mejora los de otras heurísticas, especialmente en la implementación paralela.

### **Problema VH.**

- Se estudió y definió un problema de optimización combinatoria, y se adaptó para poder ser abordado mediante técnicas evolutivas, especialmente mediante la propuesta de codificación de las posibles soluciones.
- Se propuso una técnica de búsqueda directa (SEFI) como base de experimentación para la búsqueda de soluciones aproximadas y para la configuración de poblaciones en los algoritmos evolutivos desarrollados (SEFISA y GAVH).
- Se propuso un nuevo algoritmo (GAVH) basado en combinar búsqueda directa con algoritmos genéticos, para resolver el problema. Este algoritmo ha podido ser desarrollado gracias a la experiencia adquirida en el desarrollo de otros algoritmos genéticos en dos trabajos de esta tesis: TS y ECC.
- Se sintonizaron los algoritmos y se estudió su comportamiento mediante numerosos experimentos que evaluaron sus rendimientos al compararlos con otras técnicas heurísticas propuestas para resolver este mismo problema.
- Se demostró que la propuesta de algoritmo GAVH mejora los resultados de otras heurísticas (AHP y FMADM) presentes en la literatura científica para resolver este mismo problema de optimización.
- Se adaptaron los parámetros de GAVH para contener los tiempos de ejecución en microprocesadores empujados de bajo rendimiento y consumo.

## **7.2 Trabajos futuros**

De los tres problemas de optimización abordados en esta tesis, como ya se ha comentado anteriormente, el problema VH constituye el de mayor interés, complejidad y esfuerzo dedicado. Dada la importancia actual de las comunicaciones móviles inalámbricas en redes de tecnología heterogénea, y la importancia de la satisfacción del usuario respecto a la calidad del servicio, los trabajos futuros que se plantean pertenecen a este problema de optimización.

La propuesta algorítmica GAVH desarrollada en esta tesis ha demostrado mejorar notablemente los resultados ofrecidos por algoritmos propuestos por otros autores. Este hecho, unido a las restricciones impuestas a la implementación de GAVH (rapidez en proporcionar las soluciones, bajas necesidades de recursos de memoria y procesamiento), hacen que GAVH pueda tener un gran potencial para ser propuesto en entornos reales, ya que puede ser ejecutado en la mayoría de los dispositivos móviles (*smartphones*, tabletas):

- GAVH se puede integrar fácilmente como funcionalidad en el sistema operativo *Android*, al estar programado con un código ANSI C estándar<sup>4</sup>.
- Se pueden desarrollar implementaciones de GAVH en otros lenguajes de programación para otros dispositivos móviles y sistemas operativos.

Desde el punto de vista puramente investigador, otras líneas de trabajo interesantes serían:

- Estudiar para este problema de optimización la viabilidad y, en su caso, aplicación de otras heurísticas de optimización conocidas (búsqueda tabú, evolución diferencial, algoritmos meméticos, etc.), con el propósito de mejorar la precisión de las soluciones óptimas y de minorar aún más los tiempos de computación, si fuera posible.
- Desarrollar alguna aplicación gráfica interactiva de simulación de la dinámica de dispositivos móviles sobre redes inalámbricas heterogéneas, que pueda ayudar a estudiar los algoritmos desarrollados y otras soluciones computacionales similares.

---

<sup>4</sup> Este sistema operativo soporta compiladores C, tales como C4droid, Android NDK, etc.

# 8 *Producción científica*

*“Si la sabiduría se me otorgase bajo esta condición, de mantenerla oculta y no divulgarla, la rechazaría: sin compañía no es grata la posesión de bien alguno.”*

**Lucio Anneo Séneca,**  
conocido como Séneca el Joven.  
Filósofo, político, orador y  
escritor latino del Imperio  
Romano (4 a. C. - 65 d. C.).

En total, esta tesis ha contribuido a generar 17 publicaciones científicas, cuyo grado de relación con cada línea de trabajo de tesis está explicada en los apartados titulados “Aportaciones” que hay al final de cada capítulo de esta memoria.

## 8.1 Artículos en revistas internacionales



#1

**"Embedded Intelligence for Fast QoS-based Vertical Handoff in Heterogeneous Wireless Access Networks"**. María D. Jaraíz-Simón, Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez. Pervasive and Mobile Computing, Elsevier Science, Amsterdam, The Netherlands, 2014, pp.:1-15, ISSN:1574-1192.

(Impact factor = 1.629-12)



#2

**"Fast Decision Algorithms in Low-Power Embedded Processors for Quality-of-Service Based Connectivity of Mobile Sensors in Heterogeneous Wireless Sensor Networks"**. María D. Jaraíz-Simón, Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez and Juan M. Sánchez. Sensors, Vol. 12, Issue 2, MDPI AG, Basel, Switzerland, 2012, pp.:1612-1624, ISSN:1424-8220.

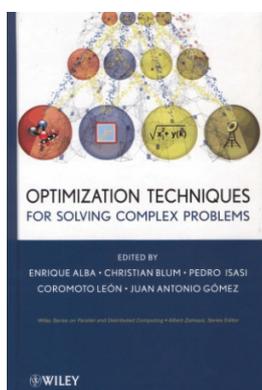
(Impact factor = 1.953)

## 8.2 Capítulos de libro



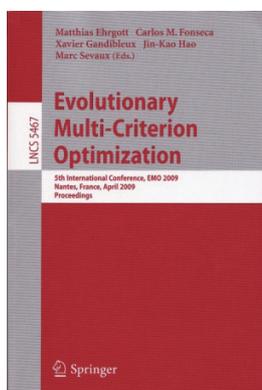
#1

"**Simulated Annealing for Real-Time Vertical-Handoff in Wireless Networks**", in: "Advances in Computational Intelligence. Lecture Notes in Computer Science, Vol. 7902". María D. Jaraíz-Simón, Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez, and Juan M. Sánchez-Pérez. I. Rojas, G. Joya, J. Gabestany (Eds.). Springer-Verlag, Berlin Heidelberg, Germany, 2013, pp.:198-209. ISBN:978-3-642-38678-7.



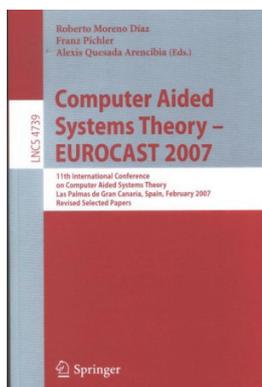
#2

"**Optimization of Time Series Using Parallel, Adaptive, and Neural Techniques. Chapter 8**", in: "Optimization Techniques for Solving Complex Problems". Juan A. Gómez, María D. Jaraíz, Miguel A. Vega-Rodríguez, and Juan M. Sánchez. John Wiley & Sons, New Jersey, USA, 2009, pp.:123-137. ISBN:978-0-470-29332-4.



#3

"**Optimizing the DFCN Broadcast Protocol with a Parallel Cooperative Strategy of Multi-Objective Evolutionary Algorithms**", in: "Evolutionary Multi-Criterion Optimization". Carlos Segura, Alejandro Cervantes, Antonio J. Nebro, María D. Jaraíz, Eduardo Segredo, Sandra García, Francisco Luna, Juan A. Gómez, Gara Miranda, Cristobal Luque, Enrique Alba, Miguel A. Vega, Coromoto León, Inés M. Galván. Springer-Verlag, Berlin Heidelberg, Germany, 2009, pp.:305-319. ISBN:978-3-642-01019-4.



#4

"**Metaheuristic Approaches for Optimal Broadcasting Design in Metropolitan MANETs**", in: "Computer Systems Aided Theory". E. Alba, A. Cervantes, J.A. Gómez, P. Isasi, M.D. Jaraíz, C. León, C. Luque, F. Luna, G. Miranda, A.J. Nebro, R. Pérez, and C. Segura. R. Moreno-Díaz et al. (Eds.), Springer-Verlag, Berlin Heidelberg, Germany, 2007, pp.:755-763. ISBN:978-3-540-75866-2.

## 8.3 Comunicaciones en congresos



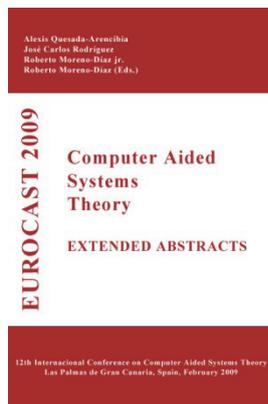
#1

**"Algoritmo Genético Empotrado en Dispositivos Móviles para Control de Conectividad en Redes Inalámbricas Heterogéneas"**. María D. Jaraíz-Simón, Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez. Actas del XV Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 13) - IX Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2013), Madrid, España, 2013, pp.:683-692. ISBN:978-84-695-8348-7.



#2

**"Procesador Dual Embebido para Vertical Handoff basado en Calidad del Servicio"**. Juan A. Gómez-Pulido, María D. Jaraíz-Simón, Miguel A. Vega-Rodríguez, Juan M. Sánchez-Pérez, Carlos T. Calafate. Actas del I Simposio en Computación Empotrada, SICE 2010, Ibergarceta Publicaciones S.L., Valencia, España, 2010, pp.:137-144. ISBN:978-84-92812-69-1.



#3

**"Parallel Differential Evolution for Multi-objective Optimization applied to the DFCN Broadcasting Strategy in MANETs"**. María D. Jaraíz-Simón, Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez, Juan M. Sánchez-Pérez. Computer Aided Systems Theory - Extended Abstracts, A. Quesada, J. Rodriguez, R. Moreno. IUCTC. Universidad de Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain, 2009, pp.:238-239. ISBN:978-84-691-8502-5.



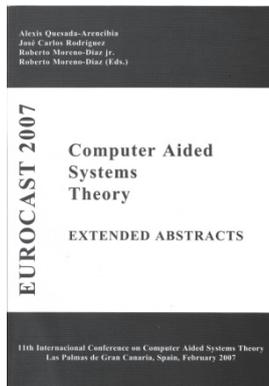
#4

**"Procesador paralelo para la medida de la calidad del diseño de códigos correctores de errores en comunicaciones"**. Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez, Juan M. Sánchez-Pérez, María D. Jaraíz-Simón. Actas de las VIII Jornadas sobre Computación Reconfigurable y Aplicaciones (JCRA 2008), José Ignacio Martínez Torre (Ed.), Univ. Rey Juan Carlos, Madrid, España, 2008, pp.:315-320. ISBN:978-84-612-5635-8.



#5

**"Algoritmo basado en Evolución Diferencial para el Problema de Redes de Radiocomunicaciones"**. J. A. Gómez Pulido, S. Mendes Priem, M. D. Jaraíz Simón, Miguel A. Vega-Rodríguez, J. M. Sánchez Pérez. Actas del V Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, F. Almeida, B. Melián, J. Moreno, M. Moreno. Universidad de La Laguna, Tenerife. España, 2007, pp.:309-314. ISBN:978-84-690-3470-5.



#6

**"Metaheuristics approaches for optimal broadcasting design in metropolitan MANETs"**. E. Alba, A. Cervantes, J.A. Gomez, P. Isasi, M.D. Jaraiz, C. Leon, C. Luque, F. Luna, G. Miranda, A. J. Nebro, R. Perez and C. Segura. Computer Aided Systems Theory - Extended Abstracts, A. Quesada, J. Rodriguez, R. Moreno. IUCTC. Universidad de Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain, 2007, pp.:262-263. ISBN:978-84-690-3603-7.



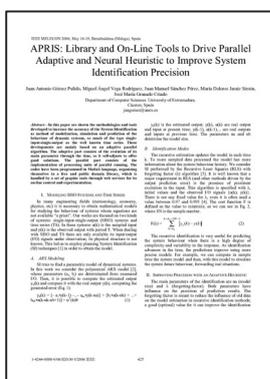
#7

**"A Differential Evolution Based Algorithm to Optimize the Radio Network Design Problem"**. Silvio Priem Mendes, Juan A. Gómez Pulido, Miguel A. Vega-Rodríguez, María D. Jaraíz Simón, Juan M. Sánchez Pérez. Proceedings of the Second IEEE International Conference on e-Science and Grid Computing, IEEE, Amsterdam, The Netherlands, 2006, pp.:119-125. ISBN:0-7695-2734-5.



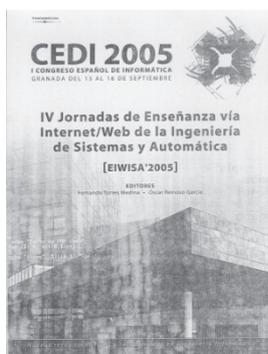
#8

**"A Genetic Algorithm to Design Error Correcting Codes"**. María D. Jaraíz Simón, Juan A. Gómez Pulido, Miguel A. Vega-Rodríguez, Juan M. Sánchez Pérez, José M. Granado Criado. Proceedings of the 13th IEEE Mediterranean Electrotechnical Conference, IEEE, Benalmádena, Málaga, Spain, 2006, pp.:807-810. ISBN:1-4244-0088-0.



#9

**"APRIS: Library and On-Line Tools to Drive Parallel Adaptive and Neural Heuristic to Improve System Identification Precision"**. Juan Antonio Gómez Pulido, Miguel A. Vega-Rodríguez, Juan Manuel Sánchez Pérez, María Dolores Jaraíz Simón, José María Granado Criado. Proceedings of the 13th IEEE Mediterranean Electrotechnical Conference, IEEE, Benalmádena, Málaga, Spain, 2006, pp.:425-428. ISBN:1-4244-0088-0.



#10

**"MOSET y TIDESI: Servicios web para el estudio de la identificación de sistemas aplicada a series temporales"**. Juan Antonio Gómez, Juan Manuel Sánchez, Miguel A. Vega-Rodríguez, María Dolores Jaraíz. Actas de las IV Jornadas de Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática, Thomson-Paraninfo, Granada, Spain, 2005, pp.:101-106. ISBN:84-9732-451-X.



#11

**"Un algoritmo genético paralelo para el aumento de la precisión de los modelos de predicción de series temporales"**. Juan Antonio Gómez, Miguel A. Vega-Rodríguez, Francisco Fernández, Juan Manuel Sánchez, María Dolores Jaraíz. Actas del IV Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, Thomson-Paraninfo, Granada, Spain, 2005, pp.:889-894. ISBN:84-9732-467-6.

# 9 Referencias

*“Dale valor a las cosas,  
no por lo que valen,  
sino por lo que significan.”*

**Gabriel García Márquez.**  
Escritor y periodista (1927 -2014 ).

- [Abbass 2002] Abbass, Hussein A., “The Self-Adaptive Pareto Differential Evolution Algorithm”. In Congress on Evolutionary Computation (CEC’2002), IEEE, volume 1, pages 831–836, May 2002.
- [Akyildiz 2005] Akyildiz, I., Wang, X., Wang, W. "Wireless mesh networks: a survey". Computer Networks , volume 47 , issue 4, pages: 445-487, 2005.
- [Alba 2004 a] Alba, E.; Chicano, J.F. “Solving the Error Correcting Code Problem with Parallel Hybrid Heuristics”, in: Proceedings of the 2004 ACM Symposium on Applied Computing: ACM press, 2004, pp. 985 – 989
- [Alba 2004 b] Alba, .E, Chicano, J. F., Dorronsoro, B., Luque, G. "Diseño de Códigos Correctores de Errores con Algoritmos Genéticos", Actas del Tercer Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB’04), Córdoba, pp. 51-58, 2004.
- [Alba 2005] Alba, E., Dorronsoro, B., Luna, F, Bouvry, P. “A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan MANETs”. Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (PDPS’05). 2005.
- [Alba 2007 a] Alba E., Cervantes, A., Gómez, J.A., Isasi, P., Jaraíz, M.D., León, C., Luque, C., Luna, F., Miranda, G., Nebro, A.J., Pérez, R., Segura, C. "Metaheuristic Approaches for Optimal Broadcasting Design in Metropolitan MANETs", in: Computer Systems Aided Theory, Springer-Verlag, 2007, pp. 755-763.
- [Alba 2007 b] Alba E., Cervantes, A., Gómez, J.A., Isasi, P., Jaraíz, M.D., León, C., Luque, C., Luna, F., Miranda, G., Nebro, A.J., Pérez, R., Segura, C. "Metaheuristics approaches for optimal broadcasting design in metropolitan MANETs", in: Computer Aided Systems Theory - Extended Abstracts, IUCTC. Univ. de Las Palmas de Gran Canaria, Spain, 2007, 262-263.
- [Anastasi 2003] Anastasi, G., Borgia, E., Conti, M., Gregori, E. “IEEE 802.11 ad hoc networks: performance measurements”. Proceedings. 23rd International Conference on Distributed Computing Systems Workshops. Pages: 758 – 763, 2003.
- [Atheros 2001] Atheros Communications, Inc., “Measured Performance of 5-GHz 802.11a Wireless LAN Systems”, White Paper, August 2001.
- [Bangnan 2003] Bangnan Xu, Hischke, S., Walke, B., ”The role of ad hoc networking in future wireless communications”.

- Communication Technology Proceedings, 2003. ICCT 2003. Int. Conference on. (2003) vol 2: 1353 – 1358.
- [Betancur 2006] Betancur, L., Hincapie, R., Bustamante, R., “WiMAX Channel - PHY Model in Network Simulator 2”, WNS2 '06: Proceeding from the 2006 workshop on ns-2: the IP network simulator 2006.
- [Bhuvaneswari 2012] Bhuvaneswari A and Raj G. 2012. An Overview of Vertical Handoff Decision Making Algorithms. *Int. J. of Computer Network and Information Security* 9, pp:55-62.
- [Bodnar 2009] Bodnar, M.R.; Curt, P.F.; Ortiz, F.E.; Carrano, C.J.; Kelmelis, E.J. An Embedded Processor for Real-Time Atmospheric Compensation. In *Proceedings of the Visual Information Processing XVIII*, Orlando, FL, USA, 14–15 April 2009; volume 7341.
- [Bruno 2005] Bruno, R., Conti, M., Gregori, E. “Mesh Networks: Commodity Multihop Ad Hoc Networks”, *Topics in Ad Hoc Networks. Communications Magazine, IEEE*. Volume:43, Issue: 3. Page(s): 123 – 131. 2005. ISSN: 0163-6804.
- [Buell 2007] Buell, D.; El-Ghazawi, T.; Gaj, K.; Kindratenko, V. High-performance reconfigurable computing. *Computer*, volume 40, 2007, 23-27.
- [Ceken 2010] Ceken C, Yarkan S, and Arslan H. 2010. Interference aware vertical handoff decision algorithm for quality of service support in wireless heterogeneous networks. *Computer Networks* 54, pp:726–740.
- [Cerny 1985] V. Cerny, A Thermodynamical Approach to the Travelling Salesman Problem: an efficient Simulation Algorithm, *Journal of Optimization Theory and Applications*, 1985, pp. 41-51.
- [Chakravorty 2004] R. Chakravorty, S. Banerjee, P. Rodriguez, J. Chesterfield, I. Pratt, “Performance optimizations for wireless wide-area networks: comparative study and experimental evaluation”, in *International Conference on Mobile Computing and Networking*, 2004, pp. 159-173.
- [Chi 2007] Chi C, Cai X, Hao R, and Liu F. 2007. Modeling and analysis of handover algorithms. 2007 *IEEE Global Telecommunications Conference (GLOBECOM'07)*, pp:4473–4477.
- [Chiasserini 2004] Chiasserini, C.-F., Cuomo, F., Piacentini, L., Rossi, M., Tinirello, I., Vacirca, F.: *Architectures and protocols for mobile computing applications: a reconfigurable approach*.

- Computer Networks: The Int. Journal of Computer and Telecommunications Networking 44 (2004) 545-567.
- [Coello 2002] Coello, Carlos A; Van Veldhuizen, David A.; Lamont, Gary B., "Evolutionary algorithms for solving multi-objective problems", Kluwer Academic Publishers, 2002.
- [Corne 2000] Corne, David W; Knowles, Joshua D.; Oates, Martin J., "The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization", Proceedings of the Parallel Problem Solving from Nature VI Conference, pages 839–848, Springer, 2000.
- [Deb 2000] Deb, Kalyanmoy; Agrawal, Samir; Pratab, Amrit; Meyerivan, T.A, "Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II", Proc. of the Parallel Problem Solving from Nature VI Conference, pages 849–858, Springer, 2000.
- [Demšar 2006] Demšar, J. "Statistical comparisons of classifiers over multiple data sets." Journal of Machine Learning Research 7, pp. 1-30, 2006.
- [Dontas 1990] Dontas, K.; De Jong, K. "Discovery of maximal Distance Codes Using Genetic Algorithms". Proceedings of the 2nd. International IEEE Conference on Tools for Artificial Intelligence (Herndon, VA), 1990, pp. 905-811.
- [Duranton 2012] Duranton, M.; Yehia, S.; Sutter, B.D.; Bosschere, K.D.; Cohen, A.; Falsafi, B.; Gaydadjiev, G.; Katevenis, M.; Maebe, J.; Munk, H.; et al. The HIPEAC Vision; Grant Agreement No: ICT-217068; HIPEAC Network of Excellence on High Performance and Embedded Architecture and Compilation; 2011. Available online: <http://www.hipeac.net/system/files/hipeacvision.pdf> (accessed on 2 February 2012).
- [Fogel 2003] Fogel, Gary B.; Corne, David W., "Evolutionary computation in bioinformatics", Morgan Kaufmann Publishers, 2003.
- [Fonseca 1993] Fonseca, Carlos M.; Fleming, Peter J., "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization", In Stephanie Forrest, editor, Proceedings of the Fifth International Conference on Genetic Algorithms, pages 416–423, University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers. 1993.
- [Gamal 1987] El Gamal, A.; Hemachandra, L.; Shperling, I.; Wei, V. "Using Simulated Annealing to Design Good Codes", IEEE Trans. of Information Theory, vol. 33, no. 1, 1987.

- [Goldberg 1989] Goldberg, D. E., "Genetic Algorithms in Search, Optimization & Machine Learning". Addison-Wesley, EE.UU., 1989.
- [Gómez-Pulido 2005a] Gómez-Pulido, J.A., Sánchez-Pérez, J.M., Vega-Rodríguez, M.A., Jaraíz-Simón, M.D., "MOSET y TIDESI: Servicios web para el estudio de la identificación de sistemas aplicada a series temporales", in: IV Jornadas de Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática, Thomson-Paraninfo, Granada, Spain, 2005, 101-106.
- [Gómez-Pulido 2005b] Gómez-Pulido, J.A., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M., Jaraíz-Simón, M.D. "Un algoritmo genético paralelo para el aumento de la precisión de los modelos de predicción de series temporales", in: IV Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, Thomson-Paraninfo, Granada, Spain, 2005, 889-894.
- [Gómez-Pulido 2006] Gómez-Pulido, J.A., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M., Jaraíz-Simón, M.D., Granado-Criado, J.M. "APRIS: Library and On-Line Tools to Drive Parallel Adaptive and Neural Heuristic to Improve System Identification Precision", in: 3th IEEE Mediterranean Electrotechnical Conference, IEEE, Benalmádena, Málaga, Spain, 2006, 425-428.
- [Gómez-Pulido 2007] Gómez-Pulido, J.A., Mendes-Priem, S., Jaraíz-Simón, M.D., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M. "Algoritmo basado en Evolución Diferencial para el Problema de Redes de Radiocomunicaciones", in: V Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, Universidad de La Laguna, Tenerife. España, 2007, 309-314.
- [Gómez-Pulido 2008] Gómez-Pulido, J.A., Jaraíz-Simón, M.D., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M. "Procesador paralelo para la medida de la calidad del diseño de códigos correctores de errores en comunicaciones", in: Actas de las VIII Jornadas sobre Computación Reconfigurable y Aplicaciones (JCRA 2008), Univ. Rey Juan Carlos, Madrid, España, 2008, 315-320.
- [Gómez-Pulido 2009] Gómez-Pulido, J.A., Jaraíz-Simón, M.D., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M. "Optimization of Time Series Using Parallel, Adaptive, and Neural Techniques", in: Optimization Techniques for Solving Complex Problems, Wiley, 2009, 127-140.

- [Gómez-Pulido 2010] Gómez-Pulido, J.A., Jaraíz-Simón, M.D., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M., Calafate, C. "Procesador Dual Embebido para Vertical Handoff basado en Calidad del Servicio", in: I Simposio en Computación Empotrada, SICE 2010, Valencia, España, 2010, 137-144.
- [Hajji 2004] Hajji, O.; Brisset, S.; Brochet, P., "A new tabu search method for optimization with continuous parameters" *Magnetics, IEEE Transactions*, Volume 40, Issue 2, Part 2, Page(s):1184 – 1187, March 2004.
- [Hasswa 2006] Hasswa A, Nasser N, and Hassanein H. 2006. Tramcar: a context-aware crosslayer architecture for next generation heterogeneous wireless networks. 2006 IEEE International Conference on Communications (ICC'06), pp: 240-245.
- [Hauck 2008] Hauck, S.; DeHon, A. *Reconfigurable Computing, Theory and Practice of FPGA-Based Computation*; Morgan Kaufmann: Waltham, MA, USA, 2008.
- [He 2011] He D, Chi C, Chan S, Chen C, Bu J, and Yin M. 2011. A Simple and Robust Vertical Handoff Algorithm for Heterogeneous Wireless Mobile Networks. *Wireless Personal Communications* 59(2), pp:361-373.
- [Hogie 2006] Hogie, L., Seredynski, M, Guinand, F, Bouvry, P. "A bandwidth-efficient broadcasting protocol for mobile multi-hop ad hoc networks". In 5th Int. Conference on Networking (ICN'06), pp: 71. IEEE Press, 2006.
- [Hogie 2007] Hogie, L. "Mobile Ad Hoc networks: modelling, simulation and broadcast-based applications." PhD thesis, Le Havre University and Luxembourg University. 2007.
- [Holland 1967] John H. Holland, *A Mathematical Framework for Studying Learning in a Classifier System. Evolution, Games and Learning: Models for Adaptation in Machines and Nature*, pp. 307-317, North-Holland, 1986.
- [Hu 2004] Hu, L.; Evans, D. Localization for Mobile Sensor Networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, Philadelphia, PA, USA, 26 Sep.–1 Oct. 2004; pp. 1-13.
- [Jaraíz-Simón 2006] Jaraíz-Simón, M.D., Gómez-Pulido, J.A., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M., Granado, J.M. "A Genetic Algorithm to Design Error Correcting Codes", in: 13th IEEE Mediterranean Electrotechnical Conference, IEEE, Benalmádena, Málaga, Spain, 2006, 807-810.
- [Jaraíz-Simón 2009] Jaraíz-Simón, M.D., Gómez-Pulido, J.A., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M. "Parallel Differential Evolution for Multi-objective Optimization

- applied to the DFCN Broadcasting Strategy in MANETs", in: Computer Aided Systems Theory - Extended Abstracts, IUUCTC, Universidad de Las Palmas de Gran Canaria, Spain, 2009, 238-239.
- [Jaraíz-Simón 2012] Jaraíz-Simón, M.D., Gómez-Pulido, J.A., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M.: Fast Decision Algorithms in Low-Power Embedded Processors for Quality-of-Service Based Connectivity of Mobile Sensors in Heterogeneous Wireless Sensor Networks. *Sensors* 12 (2012) 1612-1624.
- [Jaraíz-Simón 2013] Simulated Annealing for Real-Time Vertical-Handoff in Wireless Networks", in: "Advances in Computational Intelligence. Lecture Notes in Computer Science, Vol. 7902". María D. Jaraíz-Simón, Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez, and Juan M. Sánchez-Pérez. I. Rojas, G. Joya, J. Gabestany (Eds.). Springer-Verlag, Berlin Heidelberg, Germany, 2013, pp.:198-209. ISBN:978-3-642-38678-7.
- [Jaraíz-Simón 2013b] Algoritmo Genético Empotrado en Dispositivos Móviles para Control de Conectividad en Redes Inalámbricas Heterogéneas". María D. Jaraíz-Simón, Juan A. Gómez-Pulido, Miguel A. Vega-Rodríguez. *Actas del XV Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 13) - IX Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2013)*, Madrid, España, 2013, pp.:683-692. ISBN:978-84-695-8348-7
- [Jaraíz-Simón 2014] Jaraiz-Simon, M.D., Gomez-Pulido, J. A., Vega-Rodríguez, M. A. "Embedded Intelligence for Fast QoS-based Vertical Handoff in Heterogeneous Wireless Access Networks". *Pervasive and Mobile Computing*. Elsevier Science, Amsterdam, The Netherlands, pp: 1-15. 2014. doi: 10.1016/j.pmcj.2014.01.009.
- [Kassar 2008] Kassar, M., Kervella, B., Pujolle, G.: An overview of vertical handover decision strategies in heterogeneous wireless networks. *Computer Communications* 31 (2008) 2607–2620
- [Keung 2010] Keung, Y.; Li, B.; Zhang, Q. The Intrusion Detection in Mobile Sensor Network. In *Proceedings of the 11th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '10)*, Chicago, IL, USA, 20–24 September 2010; pp. 11-20.
- [Kirkpatrick 1983] Kirkpatrick, S; Gelatt, D; Vecchi, M.P. Optimization by Simulated Annealing. *Science*, 220, 1983, 671-680.

- [Koza 1992] Koza, J.R., Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992. ISBN 0-262-11170-5
- [Laarhoven 1987] Laarhoven, P. J. M.; Aarts, E.H.L., "Simulated annealing : theory and applications", Kluwer Academic Publishers, 1987.
- [Lassoued 2008] Lassoued, I., Bonnin, J.-M., Hamouda, Z.B., Belghith, A.: A Methodology for Evaluating Vertical Handoff Decision Mechanisms. Seventh International Conference on Networking (ICN 2008), Cancun, Mexico (2008) 377-384.
- [Lee 2005] Lee C, Chen L, Chen M, and Sun Y. 2005. A framework of handoffs in wireless overlay networks based on mobile IPv6. IEEE Journal on Selected Areas in Communications 23(11), pp: 2118-2128.
- [Lee 2006] Lee D, Han Y, and Hwang J. 2006. QoS-Based Vertical Handoff Decision Algorithm in Heterogeneous Systems. 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp:11-14.
- [Lee 2009] Lee S, Sriram K, Kim K, Kim Y, and Golmie N. 2009. Vertical Handoff Decision Algorithms for Providing Optimized Performance in Heterogeneous Wireless Networks. IEEE Transactions on Vehicular Technology 58(2), pp: 865-881.
- [Ljung 1991] Ljung, L. System Identification Toolbox. The Math Works Inc., 1991.
- [Ljung 1999] Ljung, L. System Identification. Prentice-Hall, London, 1999.
- [Matlab 2012] The MathWorks Inc., 2012: <http://www.mathworks.es/products/matlab>
- [Mendes 2006] Mendes-Priem, S., Gómez-Pulido, J.A., Vega-Rodríguez, M.A., Jaraíz-Simón, M.D., Sánchez-Pérez, J.M. "A Differential Evolution Based Algorithm to Optimize the Radio Network Design Problem", in: Second IEEE International Conference on e-Science and Grid Computing, IEEE, Amsterdam, Holanda, 2006, 1-6.
- [McNair 2004] J. McNair and F. Zhu, "Vertical Handoffs in Fourth-Generation Multinetwork Environments," IEEE Wireless Communications, vol. 11, pp. 8-15, 2004.
- [Mohanty 2006] Mohanty S and Akyildiz I. 2006. A cross-layer (layer 2 + 3) handoff management protocol for next-generation wireless systems. IEEE Transactions on Mobile Computing 5(10), pp:1347–1360.

- [Murray 2010] Murray, D., Dixon, M., Koziniec, T. "An experimental comparison of routing protocols in multi hop ad hoc networks". Telecommunication Networks and Applications Conf. (ATNAC), page(s): 159 - 164, 2010.
- [NASA 2011] NASA's Glenn Research Center. 2011. Available online: <http://www.nasa.gov/glenn> (accessed on 11 July 2011).
- [Nasser 2007] Nasser N, Guizani S, and Al-Masri E. 2007. Middleware vertical handoff manager: a neural network-based solution. 2007 IEEE International Conference on Communications (ICC'07), pp:5671-5676.
- [Nkansah 2007] Yaw Nkansah-Gyekye Y and Agbinya J. 2007. Vertical Handoff Decision Algorithm for UMTS-WLAN. The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007), pp:1-6.
- [Pahlavan 2000] Pahlavan K, Krishnamurthy P, Hatami A, Ylianttila M, Makela J, Pichna R, and Vallstron J. 2000. Handoff in hybrid mobile data networks. IEEE Personal Communications 7(2), pp:34-47.
- [Pan 2011] Pan, J.; Paul, S.; Jain R., "A Survey of the Research on Future Internet Architectures". IEEE Communications Magazine, July 2011, 26-36.
- [Parkum 1992] J.E. Parkum: "Recursive Identification of Time-Varying Systems", Ph.D. thesis, IMM, Technical University of Denmark, 1992.
- [Price 2004] Price, K. and Storn, R.. Sitio web de la evolución diferencial. <http://www.icsi.berkeley.edu/storn/code.html>, 2004.
- [QoS 2008] Quality of Service (QoS) concept and architecture, 3rd Generation Partnership Project (3GPP) TS 23.107 V8.0.0. ETSI: Sophia Antipolis Cedex, France, 2008.
- [Rayward 1996] Rayward-Smith, V. J.; Osman, I. H. and Smith, G.D., "Modern Heuristic Search Methods", John Wiley and Sons, 1996.
- [Rechenberg 1973] Ingo Rechenberg, Evolutionsstrategie, problemata, Vol. 15, Friedrich Frommann Verlag (Günther Holzboog KG), 1973.
- [Reeves 2003] Reeves, C. R; Rowe, J. E., "Genetic Algorithms – Principles and perspectives. A guide to GA Theory". Kluwer Academic Publisher, EE.UU., 2003.
- [Segura 2009] Segura, C., Cervantes, A., Nebro, A.J., Jaraíz, M.D., Segredo, E., García, S., Luna, F., Gómez-Pulido, J.A., Miranda, G., Luque, C., Alba, E., Vega-Rodríguez, M.A.,

- León, C., Galván, I.M. "Optimizing the DFCN Broadcast Protocol with a Parallel Cooperative Strategy of Multi-Objective Evolutionary Algorithms", in: *Evolutionary Multi-Criterion Optimization*, Springer, 2009, 305-319.
- [Sekihara 1992] Sekihara, K.; Haneishi, H.; Ohyama, N., "Details of simulated annealing algorithm to estimate parameters of multiple current dipoles using biomagnetic data", *Medical Imaging, IEEE Transactions*, Vol. 11, Issue 2, Page(s):293 – 299, June 1992.
- [Shah 2010] Shah, G.A.; Akan, O.B. Timing-based mobile sensor localization in wireless sensor and actuator networks. *Mob. Netw. Appl.* 2010, 15, 664-679.
- [Sharna 2011] Sharna S and Murshed M. 2011. Performance improvement of vertical handoff algorithms for QoS support over heterogeneous wireless networks. *Thirty-Fourth Australasian Computer Science Conference (ACSC '11)*, pp: 17-24.
- [Song 2005] Q. Song and A. Jamalipour, "A Network Selection Mechanism for Next Generation Networks," in *IEEE International Conference on Communications (ICC 2005)*. vol. 2: IEEE, 2005, pp. 1418-1422.
- [Song 2008] Q. Song and A. Jamalipour, "A quality of service negotiation-based vertical handoff decision scheme in heterogeneous wireless systems," *European Journal of Operational Research*, vol. 191, pp. 1059-1074, 2008.
- [Soderstrom 1989] T. Söderström et al., *System Identification*. Prentice-Hall, 1989.
- [Srinivas 1994] Srinivas, N; Deb., Kalyanmoy, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms", *Evolutionary Computation*, vol. 2, issue 3, pag:221–248, 1994.
- [Stevens 2008] Stevens-Navarro, E., Lin, Y., Wong, V.W.S.: An MDP-Based Vertical Handoff Decision Algorithm for Heterogeneous Wireless Networks. *IEEE Transactions on Vehicular Technology* 57 (2008) 1243-1254.
- [Storn 1995] Storn, Rainer; Price, Kenneth. "Differential evolution - a simple and efficient adaptative scheme for global optimization over continuous spaces". Technical Report TR-95- 12, *International Computer Science*, Berkeley, California, March 1995.
- [Tawil 2008] Tawil R, Pujolle G, and Salazar O. 2008. A vertical handoff decision scheme in heterogeneous wireless

- systems. 67th Vehicular Technology Conference (VTC'08), pp:2626-2630.
- [Tettamanzi 2001] Tettamanzi, Andrea; Tomassini, Marco, "Soft computing: integrating evolutionary, neural, and fuzzy system", Springer, 2001.
- [TS 2002] Sunspot Data Series. Royal Observatory of Belgium, Brussels, 2002.
- [TS 2003] Sunspot numbers. NOAA's National Geophysical Data Center, 2003.
- [TS 2004] A Database for Identification of Systems. <http://www.esat.kuleuven.ac.be/sista/daysi>
- [Vasu 2012] Vasu K, Maheshwari S, Mahapatra S, and Kumar C. 2012. QoS-aware fuzzy rule-based vertical handoff decision algorithm incorporating a new evaluation model for wireless heterogeneous networks. EURASIP Journal on Wireless Communications and Networking 2012, pp:3-22.
- [Veldhuizen 2003] Van Veldhuizen, D. A., Zydallis, J. B., Lamont, G. B. "Considerations in engineering parallel multiobjective evolutionary algorithms". IEEE Transactions on Evolutionary Computation 7(2), pp:144-173. 2003.
- [WangG 2005] Wang, G.; Cao, G.; La Porta, T; Zhang, W. Sensor Relocation in Mobile Sensor Networks. In Proceedigns of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005). Miami, FL, USA, 13–17 March 2005; Volume 4, pp: 2302-2312.
- [WangH 1999] Wang, H.J.; Katz, R.H.; Giese, J. Policy-Enabled Handoffs across Heterogeneous Wireless Networks. In Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications (WMCSA 99), IEEE Computer Society, New Orleans, LA, USA, 25–26 February 1999; p. 51.
- [Wolf 2007] Wolf, W. High-Performance Embedded Computing: Architectures, Applications, and Methodologies; Morgan Kaufman: Waltham, MA, USA, 2007.
- [Xia 2007] Xia L, Jiang L, He C. 2007. A novel fuzzy logic vertical handoff algorithm with aid of differential prediction and pre-decision method. 2007 IEEE International Conference on Communications (ICC'07), pp:5665-5670.
- [Xilinx 2011] Xilinx. Available online: <http://www.xilinx.com> (accessed on 11 July 2011).
- [Yan 2008] Yan X, Mani N, and Sekercioglu Y. 2008. A traveling distance prediction based method to minimize unnecessary

- handovers from cellular networks to WLANs. *IEEE Communications Letters* 12(1), pp:14-16.
- [Yan 2010] Yan X, Sekercioglu Y.A, and Narayanan S. 2010. A survey of vertical handover decision algorithms in Fourth Generation heterogeneous wireless networks. *Computer Networks* 54(11), pp:1848-1863.
- [Yang 2007] Yang K, Gondal I, Qiu B, and Dooley L. 2007. Combined SINR based vertical handoff algorithm for next generation heterogeneous wireless networks. 2007 IEEE Global Telecommunications Conf. (GLOBECOM'07), pp:4483-4487.
- [Yick 2008] Yick, J.; Mukherjee, B.; Ghosal, D. Wireless sensor network survey. *Comput. Netw.* 2008, 52, 2292-2330.
- [Zahran 2005] Zahran A and Liang B. 2005. Performance evaluation framework for vertical handoff algorithms in heterogeneous networks. 2005 IEEE International Conference on Communications (ICC'05), pp:173–178.
- [Zhou 2006] Zhou, X.; Petrov, P. Rapid and Low-Cost Context-Switch Through Embedded Processor Customization for Real-Time and Control Applications. In *Proceedings of the 43rd ACM IEEE Design Automation Conference*, San Francisco, CA, USA, 24–28 July 2006; pp. 352-357.
- [Zhu 2004] F. Zhu and J. McNair, "Optimizations for vertical handoff decision algorithms," in *Wireless Communications and Networking Conference (WCNC 2004)*. vol. 2: IEEE, 2004, pp. 867- 872.
- [Zitzler 1998] Zitzler, Eckart; Thiele, Lothar. "An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach", Technical Report 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, May 1998.
- [Zitzler 1999] Zitzler, E., Thiele, L. "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach". *IEEE Transactions on Evolutionary Computation*, 3(4), pp:257-271, 1999.
- [Zitzler 2002] Zitzler, Eckart; Laumanns, Marco; Thiele, Lothar, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2002.