



Diseño de Algoritmos Dirigido por Retos  
*Computación Evolutiva y Optimización Multimodal*

---

Challenge Driven Algorithm Design  
*Evolutionary Computation and Multimodal Optimization*

Diego Luis Afonso

---

La Laguna, 5 de septiembre de 2017

D. **Eduardo Manuel Segredo González**, con N.I.F. 78.564.242-Z Profesor Asociado adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor.

D. **Carlos Segura González**, con N.I.F. 78.704.244-S Investigador Asociado adscrito al Departamento de Computación del Centro de Investigación en Matemáticas (CIMAT), Guanajuato, México, como cotutor.

## C E R T I F I C A (N)

Que la presente memoria titulada:

*“Diseño de Algoritmos Dirigido por Retos - Computación Evolutiva y Optimización Multimodal”*

ha sido realizada bajo su dirección por D. **Diego Luis Afonso**, con N.I.F. 78.649.686-J.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 5 de septiembre de 2017

# Agradecimientos

Eduardo Manuel Segredo González  
Carlos Segura González  
Nereida Afonso Delgado  
Francisco José Luis Bermejo

# Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

## Resumen

*Durante las últimas décadas, el interés en la optimización multimodal ha crecido considerablemente, ya que muchos problemas de optimización del mundo real necesitan localizar varias posibles soluciones. Este interés ha sido fomentado aún más por múltiples competiciones de optimización. En este TFG se han implementado tres técnicas de optimización multimodal las cuales han sido incorporadas en un algoritmo genético: Fitness Sharing, Crowding Determinista y una variación del Crowding con Clusters, y han sido aplicadas a los problemas propuestos por la competición “CEC’2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization”, que se han seguido utilizando en sucesivas ediciones de dicha competición, hasta las más recientes organizadas en el CEC’2017 o en la conferencia sobre computación evolutiva GECCO 2017. El objetivo es buscar buenos parámetros de entrada de los algoritmos y comprender sus limitaciones.*

**Palabras clave:** Optimización multimodal, computación evolutiva, algoritmo genético, niching, cluster, fitness sharing, crowding.

## Abstract

Over the last decades, interest in Multimodal Optimization has grown considerably, since many real-world optimization problems need to find several solutions from the solution space. This interest has been promoted even more by research competitions. In this Final Degree Thesis three multimodal optimization techniques have been implemented and incorporated in a Genetic Algorithm: Fitness Sharing, Deterministic Crowding, and a Clustering Crowding variation. They have been applied to the problems proposed in the competition ‘CEC’2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization’ which have been used in successive editions of this competition, including the most recent ones organised in CEC’17 or the evolutionary computation conference GECCO 2017. The objective is to find common parameters and understand these algorithms’ limitations.

**Keywords:** *Multimodal optimization, evolutionary computation, genetic algorithm, niching, cluster, fitness sharing, crowding.*

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Optimización multimodal . . . . .	1
1.2. Computación evolutiva . . . . .	1
1.3. Concursos de optimización . . . . .	4
1.4. Objetivos . . . . .	9
<b>2. Descripción de técnicas algorítmicas desarrolladas</b>	<b>10</b>
2.1. Algoritmo Genético . . . . .	10
2.2. Métricas . . . . .	10
2.3. Técnicas de niching . . . . .	12
2.3.1. Fitness sharing . . . . .	12
2.3.2. Crowding . . . . .	14
2.3.3. Crowding con clusters . . . . .	14
<b>3. Evaluación experimental</b>	<b>18</b>
3.1. Fitness sharing . . . . .	18
3.2. Crowding determinista . . . . .	19
3.3. Clustering crowding . . . . .	20
3.4. Comparación de resultados . . . . .	20
<b>4. Conclusiones y líneas futuras</b>	<b>22</b>
<b>5. Conclusions and future lines of work</b>	<b>23</b>
<b>6. Presupuesto</b>	<b>24</b>
<b>Bibliografía</b>	<b>24</b>

# Índice de figuras

1.1. GA. Recombinación . . . . .	2
1.2. GA. Mutación . . . . .	2
1.3. ES. Recombinación discreta . . . . .	2
1.4. ES. Recombinación intermedia . . . . .	3
1.5. GP. Recombinación . . . . .	3
1.6. Representación gráfica de la función $F_1$ . . . . .	5
1.7. Representación gráfica de la función $F_2$ . . . . .	5
1.8. Representación gráfica de la función $F_3$ . . . . .	5
1.9. Representación gráfica de la función $F_4$ . . . . .	5
1.10. Representación gráfica de la función $F_5$ . . . . .	6
1.11. Representación gráfica de la función $F_6$ . . . . .	6
1.12. Representación gráfica de la función $F_7$ . . . . .	6
1.13. Representación gráfica de la función $F_8$ . . . . .	6
1.14. Representación gráfica de la función $F_9$ . . . . .	8
1.15. Representación gráfica de la función $F_{10}$ . . . . .	8
1.16. Representación gráfica de la función $F_{11}$ . . . . .	8
1.17. Representación gráfica de la función $F_{12}$ . . . . .	8
2.1. Disposición ideal del fitness sharing . . . . .	13
2.2. Disposición ideal del crowding . . . . .	14



# Índice de tablas

3.1. Comparación de mejores resultados de cada algoritmo . . . . .	20
6.1. Fitness sharing. Convergencia según tamaño de población ( $\epsilon = 0.01$ ) . . . . .	25
6.2. Fitness sharing. PR y SR según tamaño de población ( $\epsilon = 0.01$ ) . . . . .	26
6.3. Fitness sharing. Convergencia según $\sigma$ ( $\epsilon = 0.01$ ) . . . . .	27
6.4. Fitness sharing. PR y SR según $\sigma$ ( $\epsilon = 0.01$ ) . . . . .	28
6.5. Fitness sharing. Convergencia según probabilidad de mutación ( $\epsilon = 0.01$ ) . . . . .	29
6.6. Fitness sharing. PR y SR según probabilidad de mutación ( $\epsilon = 0.01$ ) . . . . .	30
6.7. Crowding Determinístico. Convergencia según tamaño de población ( $\epsilon = 0.1$ ) . . . . .	31
6.8. Crowding Determinístico. PR y SR según tamaño de población ( $\epsilon = 0.1$ ) . . . . .	32
6.9. Crowding Determinístico. Convergencia según probabilidad de mutación ( $\epsilon = 0.1$ ) . . . . .	33
6.10. Crowding Determinístico. PR y SR según probabilidad de mutación ( $\epsilon = 0.1$ ) . . . . .	34
6.11. Crowding con clusters. Convergencia según tamaño de población ( $\epsilon = 0.1$ ) . . . . .	35
6.12. Crowding con clusters. PR y SR según tamaño de población ( $\epsilon = 0.1$ ) . . . . .	36
6.13. Crowding con clusters. Convergencia según probabilidad de mutación ( $\epsilon = 0.1$ ) . . . . .	37
6.14. Crowding con clusters. PR y SR según tamaño de población ( $\epsilon = 0.1$ ) . . . . .	38

# Capítulo 1

## Introducción

Este TFG consiste en un trabajo de investigación, implementación y experimentación de varios algoritmos que serán aplicados a los problemas de optimización multimodal propuestos para la competición denominada ‘Special Session and Competition on Niching Methods for Multimodal Function Optimization’ organizada en el ámbito del 2017 IEEE Congress on Evolutionary Computation (CEC 2017).

### 1.1. Optimización multimodal

Los problemas de optimización pretenden encontrar el conjunto de valores de las variables de decisión que, en el caso de problemas de maximización, obtengan el mayor valor posible de una función objetivo. A partir de una función  $f(\vec{X})$  y un vector de variables de decisión  $\vec{X} = [x_0, x_1, \dots, x_D]$ , siendo  $D$  la dimensión del problema y cada  $x_{1\dots D}$  un número real, en el caso que nos ocupa, el valor de fitness de un vector  $\vec{X}$  equivale al valor de la función objetivo  $f(\vec{X})$  ( $f : \Omega \subseteq R^D \rightarrow R$ ). El objetivo de un problema de maximización, es encontrar el vector  $\vec{X}' \in \Omega$  para el que  $f(\vec{X}') \geq f(\vec{X})$ , para todo  $\vec{X} \in \Omega$ , siendo  $\Omega$  el espacio de decisión [3]. La aplicación de algoritmos evolutivos a este tipo de problemas se ha extendido ampliamente en las últimas décadas.

En el contexto de la optimización multimodal, las funciones con las que se trabajan cuentan con multitud de óptimos globales, por lo tanto, el objetivo es encontrar el conjunto de soluciones con mayor valor de la función objetivo. De nuevo a partir de la misma función  $f(\vec{X})$  y el vector de variables de decisión  $\vec{X} = [x_0, x_1, \dots, x_D]$ , se pretende hallar el conjunto de vectores de soluciones  $\vec{X}_1, \vec{X}_2, \dots, \vec{X}_n \subset X$  que cumplan  $f(\vec{X}_i) \geq f(\vec{X})$  con  $i = 1, \dots, n$ , para todo  $\vec{X} \in \Omega$  y que para todo  $\vec{X}_i \in X$  y  $\vec{X}_j \in X$  con  $i = 1, \dots, n, j = 1, \dots, n$  e  $i \neq j$ ,  $f(\vec{X}_i) = f(\vec{X}_j)$ .

### 1.2. Computación evolutiva

Los algoritmos elegidos han sido los algoritmos evolutivos, un conjunto de algoritmos poblacionales en los que los diferentes individuos o soluciones exploran el espacio de búsqueda asociado a una función determinada para tratar de encontrar el óptimo u óptimos globales de la misma. Estas técnicas permiten la abstracción del dominio del problema, así como fiabilidad y gran capacidad de búsqueda, además de ser sencillas de implementar en comparación con otras técnicas de búsqueda. Existen multitud de variantes de estos algoritmos, entre ellas se pueden destacar Estrategias Evolutivas [7], Programación Genética [8], Evolución Diferencial [10], entre otras. En este trabajo se usarán los Algoritmos Genéticos [4], una de las primeras variantes de algoritmos evolutivos que surgió.

Los algoritmos evolutivos están basados en la teoría de la evolución de las especies [2]. En un cierto entorno, los seres vivos de una misma especie compiten entre ellos para conseguir recursos y por el derecho a reproducirse. El nivel de adaptación de los individuos al medio define la capacidad de los mismos para sobrevivir y reproducirse, de manera que individuos mejor adaptados tengan más probabilidades de alcanzar estas metas. Usando de base estas ideas, los algoritmos evolutivos crean un conjunto de soluciones candidatas (individuos de la población) que tendrán un valor de la función objetivo asociado o valor de fitness (grado de adaptación al medio). Estos candidatos competirán

entre sí y los mejor adaptados podrán reproducirse o sobrevivir a lo largo de varias generaciones o iteraciones del algoritmo.

Los individuos pueden estar representados internamente de múltiples formas. Inicialmente se usaban cadenas bits, pero también es común el uso de vectores de números reales o estructuras de árbol. Esta representación define la genética del individuo y, a su vez, constituye los operadores con los que se realizan la recombinación y la mutación. La recombinación, también llamada reproducción, consiste en generar uno o más hijos a partir de dos o más padres y la mutación consiste en la transformación instantánea de los genes de un individuo.

Según [1], algunas de las variantes más ampliamente utilizadas que pertenecen a la familia de los algoritmos evolutivos son:

- **Algoritmos Genéticos (GA).** Son los algoritmos evolutivos más conocidos. Inicialmente utilizaban cadenas de bits para representar a los individuos de la especie, y generalmente se seleccionan los individuos mejor adaptados para la etapa de reproducción. La recombinación de soluciones (Figura 1.1) consiste en el intercambio de parte de la información genética a partir de una variable de decisión seleccionada aleatoriamente. Las mutaciones (Figura 1.2), por su parte, consisten en cambiar el valor de una variable de decisión y, generalmente, tienen pocas probabilidades de ocurrir.

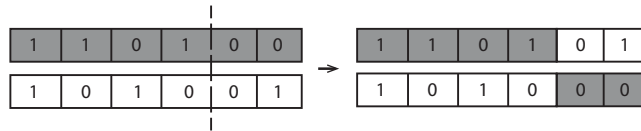


Figura 1.1: GA. Recombinación

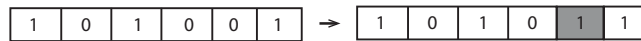


Figura 1.2: GA. Mutación

Tradicionalmente funcionan de forma fija: el tamaño de población es constante. Después del cruce aleatorio, en el algoritmo original, la nueva generación reemplaza a la anterior completamente y de forma inmediata, aunque se pueden utilizar multitud de técnicas de selección de supervivientes. A continuación se aplica la mutación a cada individuo de la nueva generación. Cabe destacar que es posible que algunos individuos permanezcan inalterados tras las operaciones de mutación y cruce y sobrevivan varias generaciones.

- **Estrategias Evolutivas (ES).** Como características generales se puede destacar que los individuos están representados mediante vectores de números reales. Inicialmente, la reproducción consistía en la adición de un número aleatorio a cada uno de los elementos del vector de forma independiente, de manera que el nuevo individuo se aceptaría únicamente si su fitness es mejor. Esquemas posteriores utilizan operaciones de recombinación y cruce y los más usados sustituyen a todos los padres en cada generación. En la recombinación, generalmente se utilizan dos padres para crear un único hijo. Los dos tipos de recombinación más utilizados son la discreta (Figura 1.3), en la que cada alelo es elegido de uno de los padres de forma aleatoria, y la intermedia (Figura 1.4), que consiste en calcular la media de los alelos de los padres:

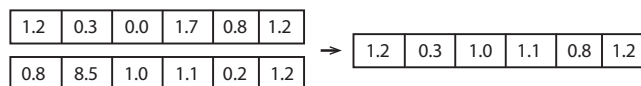


Figura 1.3: ES. Recombinación discreta

- **Programación Evolutiva (EP).** Fue desarrollada para simular la evolución como un proceso de aprendizaje con el objetivo de generar inteligencia artificial, entendiendo inteligencia como la capacidad del sistema de adaptar su comportamiento para cumplir una serie de objetivos

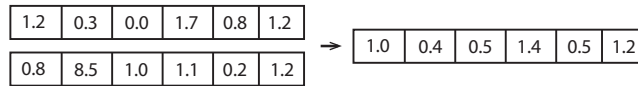


Figura 1.4: ES. Recombinación intermedia

*específicos en un rango de entornos* [1]. Originalmente los individuos se representaban como máquinas con un conjunto de estados finito. Actualmente se tiende a utilizar números reales. No se utiliza recombinação porque se considera que los individuos pertenecen a especies diferentes. Cada progenitor genera un descendiente con el que luego deberá competir para decidir quién sobrevive. Esto implica que el único operador de variación es la *mutación*, que consistiría en la adición de un número aleatorio al número que representa al individuo.

Este tema ha sido discutido de forma intensa y, se han realizado comparaciones de EP con y sin cruce. Según los resultados de [5], en la EP la capacidad de la mutación para generar descendencia mejor adaptada es superior a la de la recombinação en las primeras etapas del algoritmo, mientras que la recombinação genera mejores resultados en etapas posteriores.

- Programación Genética (GP).** Son una de las técnicas más modernas de los algoritmos evolutivos y se suelen utilizar en el campo del aprendizaje automático. En GP se utilizan estructuras de árbol como cromosomas que representan expresiones con una sintaxis determinada, por ejemplo, expresiones aritméticas, o código de un determinado lenguaje de programación. La recombinação (Figura 1.5) consiste en el intercambio de sub-árboles entre soluciones, de manera que dos ascendientes generan dos descendientes. La mutación conlleva cambios aleatorios en las hojas del árbol o en sub-árboles intermedios. La elección de los progenitores depende del grado de adaptación de los mismos, y por norma general, cada generación sustituye completamente a la anterior. Hay que tener en cuenta que la reproducción conlleva recombinação o mutación, pero nunca ambas juntas, de hecho, hay muchos casos en los que no se utiliza ningún tipo de mutación. La inicialización puede realizarse de múltiples maneras. La más utilizada se llama

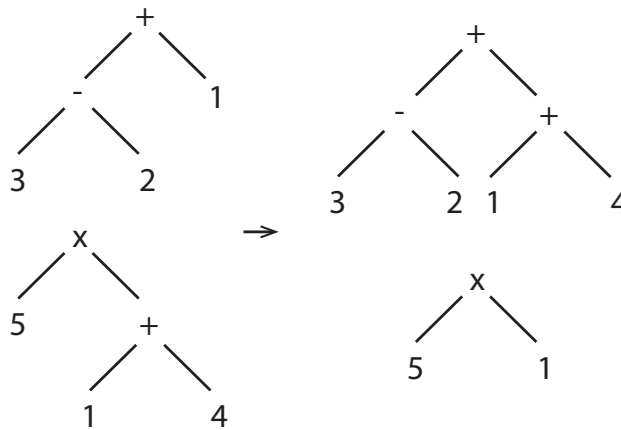


Figura 1.5: GP. Recombinación

método **ramped half-and-half**. Primero se define una profundidad máxima para los árboles y luego la población inicial se crea a partir de un conjunto de funciones (F) y otro de terminales (T). Generalmente se utilizan dos estrategias que tendrán la misma probabilidad de ser seleccionadas, la *completa*, en el que todas las ramas tendrán profundidad máxima, y la *creciente*, en la que las ramas pueden tener diferentes profundidades. En ambos casos, el contenido de los árboles se selecciona de los conjuntos F y T de forma aleatoria.

- Evolución Diferencial (DE).** Estas técnicas adquieren su nombre de las operaciones de mutación que utilizan: la **mutación diferencial**. Dado un vector solución ( $\vec{x}$ ), una mutación del mismo ( $\vec{x}'$ ) se obtiene añadiéndole un "vector de perturbación" ( $\vec{p}$ ), que se obtiene a través del producto escalar de la diferencia de dos vectores de la población escogidos de forma aleatoria ( $\vec{y}$ )

y  $\vec{z}$ ):

$$\begin{aligned}\vec{x}' &= \vec{x} + \vec{p} \\ \vec{p} &= F.(\vec{y} - \vec{z})\end{aligned}$$

El factor de escala  $F$  se usa para controlar el ritmo de evolución de la población. La población  $P = \{\vec{x}_1, \dots, \vec{x}_i, \dots, \vec{x}_\mu\}$  está constituida por una lista de individuos, de manera que se puede acceder a ellos por su índice. La ejecución de una generación consiste en la creación de un vector de mutación  $M = \{\vec{m}_1, \dots, \vec{m}_i, \dots, \vec{m}_\mu\}$ . Aunque existen diversas estrategias de mutación, teniendo en cuenta las más frecuentemente utilizadas, cada uno de los mutantes se genera mediante las fórmulas anteriores a partir de tres vectores de la población escogidos de manera aleatoria. A continuación se crea un nuevo vector  $T = \{\vec{t}_1, \dots, \vec{t}_i, \dots, \vec{t}_\mu\}$ , donde cada  $\vec{t}_i$  es el resultado de aplicar, típicamente, un operador de cruce binomial a  $\vec{x}_i$  y  $\vec{m}_i$ . Cada par  $\vec{x}_i$  y  $\vec{t}_i$  compiten en un torneo binario y el ganador sobrevive la siguiente generación.

- **Algoritmos de Estimación de Distribución (EDA).** Los EDA buscan estimar la **distribución de probabilidad** de cada variable de decisión asociada al problema. En ellos las operaciones de recombinación y mutación son completamente diferentes. Primero se crea un "modelo gráfico" que representa las dependencias entre las variables que definen una posible solución. A continuación se estiman los parámetros del modelo para crear una distribución de probabilidad condicional sobre las variables. Finalmente se genera la descendencia muestreando esta distribución. Es decir, la nueva generación se crea a partir de la distribución de probabilidad obtenida de los mejores individuos de la generación anterior.

Estas técnicas evolutivas suelen converger rápidamente hacia un único óptimo de la función. Este no es el objetivo de la optimización multimodal, ya que se pretenden encontrar todos los óptimos de cada función, por ello, será necesario investigar e implementar las conocidas como *técnicas de niching*, que pretenden evitar esta convergencia única y promueven la generación de *nichos* o cúmulos de individuos en los alrededores de los óptimos locales/globales de la función.

### 1.3. Concursos de optimización

Actualmente competiciones como la que se abarcará en este TFG, que fomentan la investigación en el campo de la optimización en general, y de la optimización multimodal en particular, son muy comunes, ya que muchos problemas reales de ámbitos muy diferentes pueden ser formulados como problemas de optimización y en muchos casos se requiere la localización de varias posibles soluciones. Algunos ejemplos de estas competiciones son el 'GECCO 2017' donde se proponía la resolución de los mismos problemas que se abarcarán en este TFG, el 'Global Trajectory Optimisation Competition', la 'International Conference on Evolutionary Multi-Criterion Optimization' o el 'Generalization-based Contest in Global Optimization'.

La competición particular en la que se basa este TFG presenta un total de veinte funciones con diferentes dimensiones a las que se le deben aplicar las técnicas que se consideren oportunas para lograr encontrar todos sus óptimos globales. Las funciones presentadas son las siguientes [9]:

$F_1$  (Figura 1.6) tiene dos óptimos globales y está definida por la siguiente función:

$$f_1(x) = \begin{cases} 80(2,5 - x), & \text{for } 0 \leq x < 2,5 \\ 64(x - 2,5), & \text{for } 2,5 \leq x < 5,0 \\ 64(7,5 - x), & \text{for } 5,0 \leq x < 7,5 \\ 28(x - 7,5), & \text{for } 7,5 \leq x < 12,5 \\ 28(17,5 - x), & \text{for } 12,5 \leq x < 17,5 \\ 32(x - 17,5), & \text{for } 17,5 \leq x < 22,5 \\ 32(27,5 - x), & \text{for } 22,5 \leq x < 27,5 \\ 80(x - 27,5), & \text{for } 27,5 \leq x < 30 \end{cases}$$

$F_2$  (Figura 1.7) tiene cinco óptimos globales y está definida por la siguiente función:

$$f_2(x) = \sin^6(5\pi x)$$

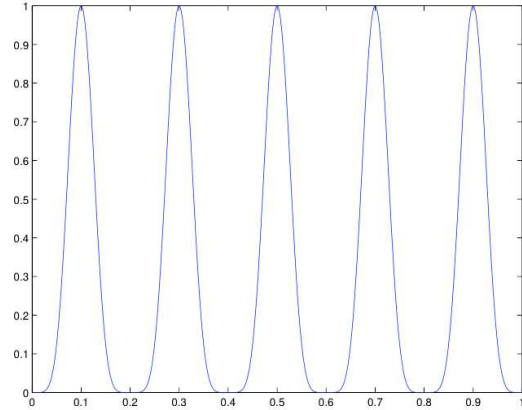
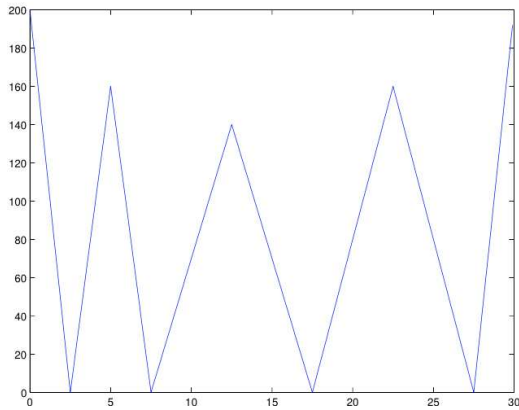


Figura 1.6: Representación gráfica de la función  $F_1$

Figura 1.7: Representación gráfica de la función  $F_2$

$F_3$  (Figura 1.8) tiene un óptimo global y está definida por la siguiente función:

$$f_3(x) = \exp\left(-2\log(2)\left(\frac{x-0,08}{0,854}\right)^2\right) \sin^6\left(5\pi\left(x^{\frac{3}{4}}-0,05\right)\right)$$

$F_4$  (Figura 1.9) tiene cuatro óptimos globales y está definida por la siguiente función:

$$f_4(x, y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7^2)$$

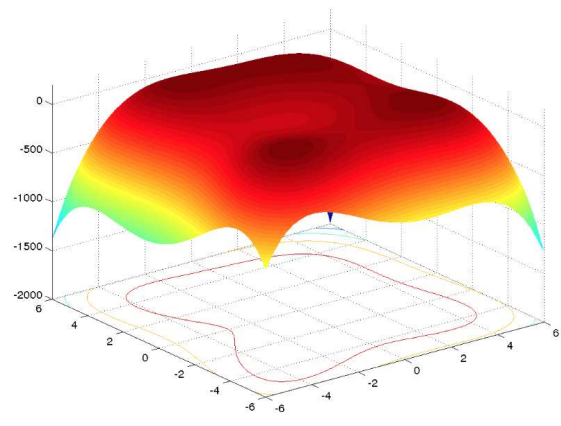
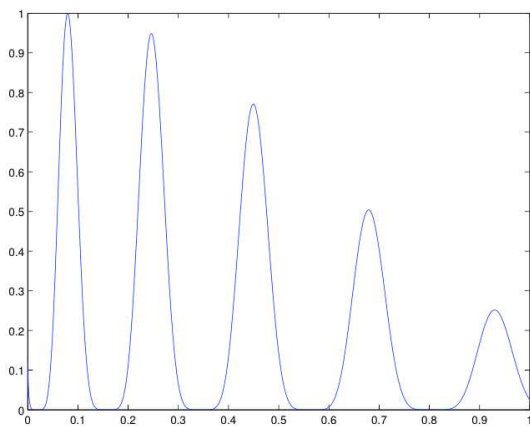


Figura 1.8: Representación gráfica de la función  $F_3$

Figura 1.9: Representación gráfica de la función  $F_4$

$F_5$  (Figura 1.10) tiene dos óptimos globales y está definida por la siguiente función:

$$f_5(x, y) = -4 \left[ \left( 4 - 2,1x^2 + \frac{x^4}{3} \right) x^2 + xy + (4y^2 - 4) y^2 \right]$$

$F_6$  (Figura 1.11) tiene un total de  $D \cdot 3^D$  óptimos globales, debe estudiarse con  $D = 2$  y con  $D = 3$  y está definida por la siguiente función:

$$f_6(\vec{x}) = - \prod_{i=1}^D \sum_{j=1}^5 j \cos[(j+1)x_i + j]$$

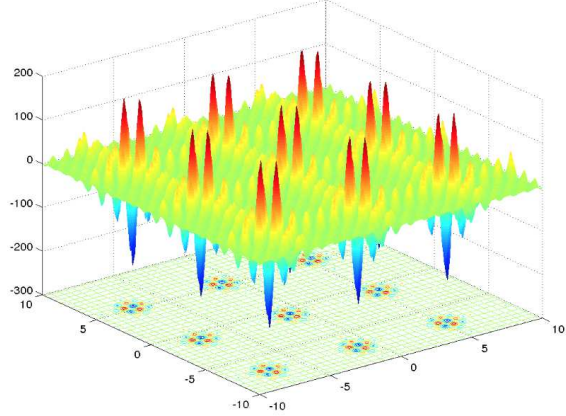
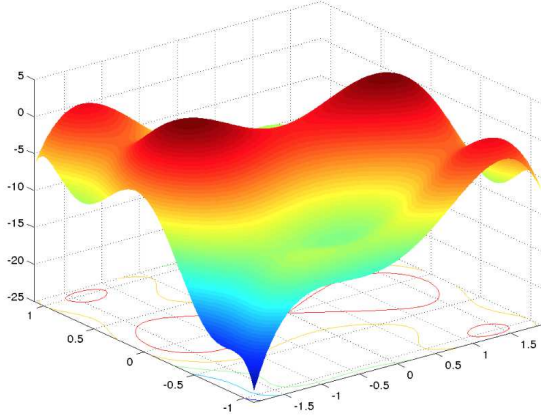


Figura 1.10: Representación gráfica de la función  $F_5$

Figura 1.11: Representación gráfica de la función  $F_6$

$F_7$  (Figura 1.12) tiene un total de  $6^D$  óptimos globales, debe estudiarse con  $D = 2$  y con  $D = 3$  y está definida por la siguiente función:

$$f_7(\vec{x}) = \frac{1}{D} \sum_{i=1}^D \sin(10 \log x_i)$$

$F_8$  (Figura 1.13) tiene un total de  $\prod_{i=1}^D k_i$  óptimos globales, debe estudiarse con  $D = 2$  y está definida por la siguiente función:

$$f_8(\vec{x}) = - \sum_{i=1}^D (10 + 9 \cos(2\pi k_i x_i))$$

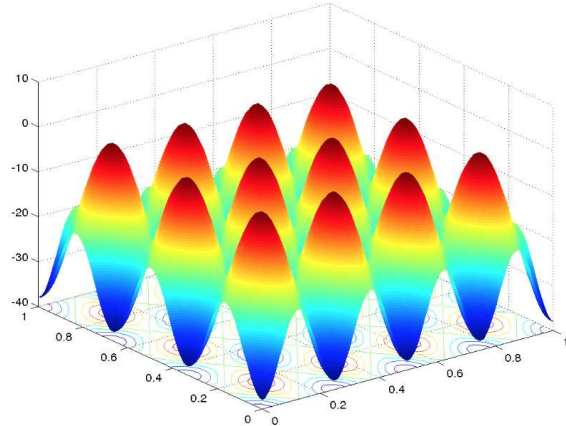
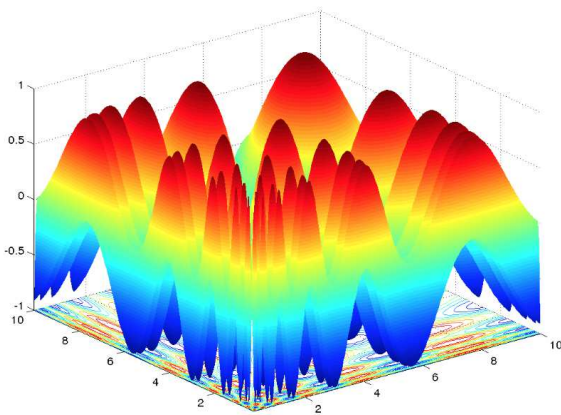


Figura 1.12: Representación gráfica de la función  $F_7$

Figura 1.13: Representación gráfica de la función  $F_8$

Para definir las últimas tres funciones, es necesario comprender cómo se construyen funciones multimodales compuestas. Una función compuesta de  $D$  dimensiones  $CF_j : A_D \subset \mathbb{R}^D \rightarrow \mathbb{R}$  se construye de forma general como la agregación ponderada de  $n$  funciones básicas  $f_i : A_D \subset \mathbb{R}^D \rightarrow \mathbb{R}$ . Las

funciones básicas se desplazan en el espacio de la función  $A_D$  y pueden usarse directamente o rotarse mediante matrices de transformación. De esta forma, una función compuesta  $CF_j$  se calcula mediante la ecuación siguiente:

$$CF_j(\vec{x}) = \sum_{i=1}^n w_i \left( \hat{f}_i((\vec{x} - \vec{o})/\lambda_i \cdot M_i) + bias_i \right) + f_{bias'}^j$$

Siendo  $n$  el número de funciones básicas usadas,  $\hat{f}_i$  la normalización de la función  $i$ -ésima,  $i \in 1, 2, \dots, n$ ,  $w_i$  el peso,  $\vec{o}_i$  el nuevo óptimo desplazado de cada  $\hat{f}_i$ ,  $M_i$  la matriz de transformación de cada  $\hat{f}_i$ , y  $\lambda_i$  se utiliza para comprimir o dilatar la función  $\hat{f}_i$ . Además se incluyen dos parámetros,  $bias_i$ , que determina qué óptimo será el global, y  $f_{bias'}^j$ , que define un valor de sesgo para función de composición. Para que el óptimo global de cada función básica constituya un óptimo global de la función compuesta, se fija  $bias_i = 0$ ,  $i \in 1, 2, \dots, n$ , además de  $f_{bias'}^j = 0$ , para que el fitness de todos los óptimos globales de la función final sean cero.

El peso  $w_i$  de cada función básica se calcula con la siguiente ecuación:

$$w_i = \exp \left( - \frac{\sum_{k=1}^D (x_k - o_{ik})^2}{2D\sigma_i^2} \right)$$

$$w_i = \begin{cases} w_i, & w_i = \max(w_i) \\ w_i(1 - \max(w_i)^{10}), & \text{en otro caso} \end{cases}$$

Los pesos se normalizan siguiendo  $w_i = w_i / \sum_{i=1}^n w_i$  y  $\sigma_i$  controla el rango de cobertura de cada función básica.

Las funciones básicas utilizadas son las siguientes:

- Función esfera:

$$f_S(\vec{x}) = \sum_{i=1}^D x_i^2$$

- Función de Griewank:

$$f_G(\vec{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$$

- Función de Rastrigin:

$$f_R(\vec{x}) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

- Función Weierstrass:

$$f_W(\vec{x}) = \sum_{i=1}^D \left( \sum_{k=0}^{kmax} \alpha^k \cos(2\pi \beta^k (x_i + 0,5)) \right) - D \sum_{k=0}^{kmax} \alpha^k \cos(2\pi \beta^k (0,5))$$

siendo  $\alpha = 0,5$ ,  $\beta = 3$  y  $kmax = 20$ .

- Función de Griewank expandida más función de Rosenbrock (EF8F2):

$$F8(\vec{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$$

$$F2(\vec{x}) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$$

$$EF8F2(\vec{x}) = F8F2(x_1, x_2, \dots, x_D)$$

$$= F8(F2(x_1, x_2)) + F8(F2(x_2, x_3)) + \dots$$

$$+ F8(F2(x_{D-1}, x_D)) + F8(F2(x_D, x_1))$$



A partir de todas estas funciones básicas se definen el resto de problemas.

$F_9$  está compuesta por un total de seis funciones:  $f_1$  y  $f_2$  son funciones de Griewank,  $f_3$  y  $f_4$  son funciones Weierstrass y  $f_5$  y  $f_6$  son funciones esfera.. Tiene un total de seis óptimos en el espacio  $A_D = [-5, 5]^D$  y debe estudiarse con  $D = 2$ .

$F_{10}$  está compuesta por ocho funciones básicas:  $f_1$  y  $f_2$  son funciones de Rastrigin,  $f_3$  y  $f_4$  son funciones Weierstrass,  $f_5$  y  $f_6$  son funciones de Griewank y  $f_7$  y  $f_8$  son funciones esfera. Cuenta con ocho óptimos en el espacio  $A_D = [-5, 5]^D$  y debe estudiarse con  $D = 2$ .

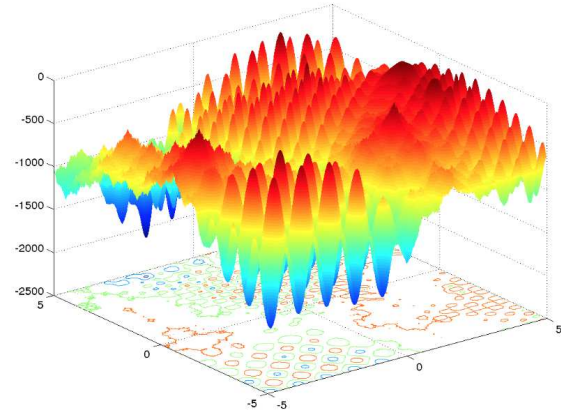
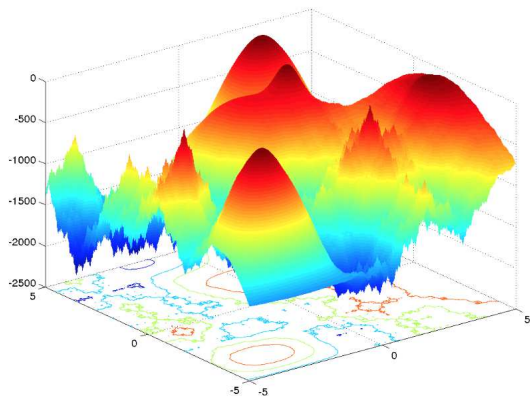


Figura 1.14: Representación gráfica de la función  $F_9$

Figura 1.15: Representación gráfica de la función  $F_{10}$

$F_{11}$  está compuesta por seis funciones básicas:  $f_1$  y  $f_2$  son funciones de EF8F2,  $f_3$  y  $f_4$  son funciones Weierstrass, y  $f_5$  y  $f_6$  son funciones de Griewank. Cuenta con seis óptimos en el espacio  $A_D = [-5, 5]^D$  y debe estudiarse con  $D = 2$ ,  $D = 3$ ,  $D = 5$ ,  $D = 10$ .

$F_{12}$  está compuesta por ocho funciones básicas:  $f_1$  y  $f_2$  son funciones de Rastrigin,  $f_3$  y  $f_4$  son funciones EF8F2,  $f_5$  y  $f_6$  son funciones de Weierstrass y  $f_7$  y  $f_8$  son funciones Griewank. Cuenta con ocho óptimos en el espacio  $A_D = [-5, 5]^D$  y debe estudiarse con  $D = 3$ ,  $D = 5$ ,  $D = 10$  y  $D = 20$ .

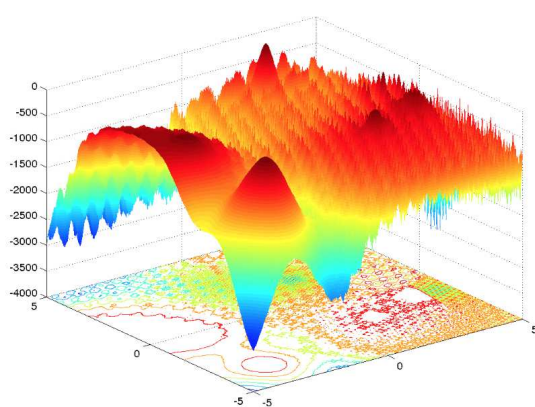
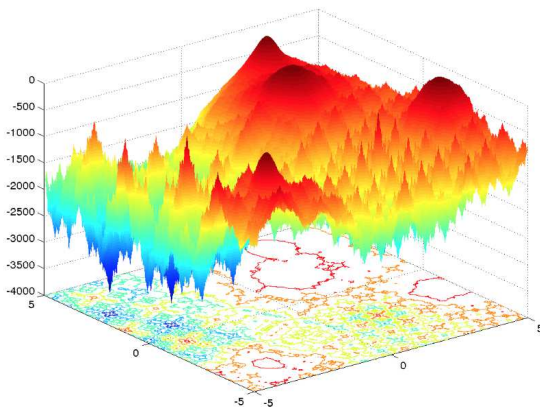


Figura 1.16: Representación gráfica de la función  $F_{11}$

Figura 1.17: Representación gráfica de la función  $F_{12}$

Más información sobre las funciones y la competición en general pueden encontrarse en el enunciado de la misma<sup>1</sup>

<sup>1</sup><http://www.cec2017.org/>

## **1.4. Objetivos**

Con este trabajo se pretende comprender la importancia y complejidad de la optimización multimodal y familiarizar al alumno con las estrategias y los algoritmos que se utilizan para resolver problemas de este tipo. Además, también se busca introducir al alumno en el mundo de la investigación computacional mediante la realización de tareas de búsqueda de información en artículos y libros de este ámbito, de implementación y experimentación con algoritmos de búsqueda adaptados a este tipo de problemas, y de redacción y exposición de la información recopilada y las conclusiones alcanzadas. Todo ello enmarcado en el ámbito de una competición que sirva de motivación.

## Capítulo 2

# Descripción de técnicas algorítmicas desarrolladas

### 2.1. Algoritmo Genético

Los algoritmos implementados se han basado en un algoritmo genético que ha sido modificado mediante la incorporación de conceptos de diferentes técnicas de niching. El Pseudocódigo 1 describe su funcionamiento a alto nivel, pero los pasos que se realizan son los siguientes:

1. Se parte de una población repartida de manera uniforme por el espacio.
2. Se realiza la recombinación de la población. Por norma general, se genera una cantidad de descendientes igual al número de padres.
3. Se les aplica la operación de mutación a los hijos.
4. Se calcula el valor de fitness de los individuos de la nueva generación.
5. Se eligen los supervivientes dependiendo la técnica de selección elegida.
6. Se repite desde el segundo hasta que se alcance el número máximo de evaluaciones.

Cabe destacar que los individuos están formados por vectores de números reales que constituyen a su vez las coordenadas en el espacio de la función. La recombinación consiste en el cruce de los vectores de dos padres en un punto elegido al azar. La mutación, por su parte, consiste en la adición de un número aleatorio a cada posición del vector.

Como parámetros de entrada del algoritmo, es necesario fijar el tamaño de la población, las probabilidades de que se realicen las operaciones de mutación y cruce, la estrategia de selección de supervivientes y la técnica de niching a emplear. Se puede elegir entre tres tipos de selección de supervivientes: *steady state*, que genera un único hijo en cada generación y sustituye a un padre; generacional, que sustituye la población completa por la nueva generación; o la selección de los mejores individuos entre padres e hijos. Hay que tener en cuenta que en algunas de las modificaciones el parámetro que define la estrategia de selección de supervivientes se ignora porque se emplea otra estrategia propia del algoritmo de niching.

Las técnicas de niching implementadas se comentan en un próximo apartado de este mismo capítulo. En particular, se han implementado tres técnicas diferentes: fitness sharing, deterministic crowding y un algoritmo propuesto.

### 2.2. Métricas

La competición propone una serie de métricas que determinan de manera inequívoca la calidad de los resultados obtenidos tras la ejecución de los algoritmos. Para ello es necesario implementar

---

**Algoritmo 1:** Algoritmo Genético

---

**Input :**  $np$  - tamaño de la población  
 $pm$  - probabilidad de mutación  
 $pc$  - probabilidad de cruce  
 $ss$  - método de selección de supervivientes  $mEval$  - número máximo de evaluaciones

```

1 P =  $\emptyset$ 
2 Inicialización uniforme de P con np individuos
3 while  $mEval > eval$  do
4   Off =  $\emptyset$ 
5   Off  $\leftarrow$  P.cruzar(pc);
6   Off = Off.mutar(pm);
7   Off.evaluar();
8   P = selecciónSupervivientes(ss);
9   Off =  $\emptyset$ 
10 end

```

---



---

**Algoritmo 2:** Comprobación de óptimos alcanzados

---

**Input :**  $L_{sorted}$  - lista de individuos ordenada de forma decreciente según del valor de fitness.  
 $\epsilon$  - nivel de precisión  
 $r$  - radio del nicho  
 $ph$  - valor de fitness del óptimo global  
(r y ph tienen valores fijados para cada problema)

**Output:** S - lista de individuos indentificados como soluciones

```

1 S =  $\emptyset$ 
2 for p en  $L_{sorted}$  do
3   encontrado  $\leftarrow$  false;
4   if  $dist(ph, fitness(p)) \leq \epsilon$  then
5     for todo s  $\in$  S do
6       if  $(dist(s, p) \leq r)$  then
7         encontrado  $\leftarrow$  true;
8         break;
9       end
10      if not found then
11        S  $\leftarrow$  S  $\cup$  {p};
12      end
13    end
14  end
15 end

```

---

un pequeño método que será ejecutado cada cierto número de evaluaciones y comprobará si se han alcanzado todos los óptimos globales de la función.

Como puede apreciarse en el Pseudocódigo 2, el algoritmo crea una lista de aquellos individuos que se encuentran próximos a los óptimos globales de la función. A partir de esta información se puede determinar la calidad de las soluciones obtenidas en una ejecución determinada mediante el cálculo de tres métricas diferentes:

- **Peak Ratio (PR):** Proporción de óptimos globales encontrados con respecto al total. Se calcula mediante la siguiente fórmula:

$$PR = \frac{\sum_{i=1}^{NR} NPF_i}{NKP * NR}$$

Siendo  $NPF_i$  el número de óptimos globales encontrados en la  $i$ -ésima ejecución,  $NKP$  el número de óptimos conocidos y  $NR$  el número de ejecuciones.

- **Successful Runs (SR):** Proporción de ejecuciones exitosas, considerándose exitosa aquella ejecución en la que se localizan todos los óptimos globales. Este valor se calcula usando la siguiente fórmula:

$$SR = \frac{NSR}{NR}$$

Donde  $NSR$  es el número de ejecuciones exitosas.

- **Convergence (Conv):** Media de evaluaciones requeridas para localizar todos los óptimos globales. El cálculo se realiza mediante la siguiente función:

$$Conv = \frac{\sum_{i=1}^{NR} FE_i}{NR}$$

Siendo  $FE_i$  el número de evaluaciones empleadas en la ejecución  $i$ .

## 2.3. Técnicas de niching

Como se ha comentado anteriormente, en los problemas de optimización multimodal, se trabaja con funciones que cuentan con multitud de óptimos globales/locales, y el objetivo es encontrarlos todos. Las técnicas de niching pretenden evitar que los algoritmos converjan hacia un sólo óptimo global, haciendo que se formen grupos de individuos en los alrededores de los óptimos locales/globales. Se han implementado tres métodos de niching diferentes para resolver los problemas de la competición. Dentro de las técnicas tradicionales de niching podemos encontrar el *fitness sharing* y el *crowding* y, aunque no crean los nichos de forma explícita, utilizan técnicas para intentar promover que los individuos se agrupen formando dichos nichos. El tercer método implementado es una variación del *crowding*, en la que se han agrupado a los individuos en clusters con los que se puede interactuar.

### 2.3.1. Fitness sharing

El *fitness sharing* constituye la técnica de niching tradicional que se debía implementar. Su funcionamiento es sencillo: se define una distancia de radio de nicho  $\sigma$ . Si dos o más individuos se encuentran a una distancia menor que  $\sigma$ , su *fitness* se penalizará. La penalización será mayor cuantos más individuos haya en el nicho y cuanto más cerca se encuentren entre sí. La función que define la penalización es la siguiente:

$$f'_i = \frac{f_i}{m_i}$$

$$m_i = \sum_{j=1}^N sh(d_{ij})$$

Siendo  $d_{ij}$  la distancia entre los individuos  $i$  y  $j$  y sabiendo que  $sh(d_{ij})$  esta definido por la siguiente función:

$$sh(d_{ij}) = \begin{cases} 1 - (d_{ij}/\sigma), & \text{si } d < \sigma \\ 0, & \text{en otro caso} \end{cases}$$

Al realizar esta penalización, se consigue que el valor de la función objetivo de individuos localizados en óptimos locales muy concurridos sea menor, y se promueve la expansión hasta óptimos no visitados. La disposición de los individuos que se desea alcanzar es similar a la que aparece en la Figura 2.1.

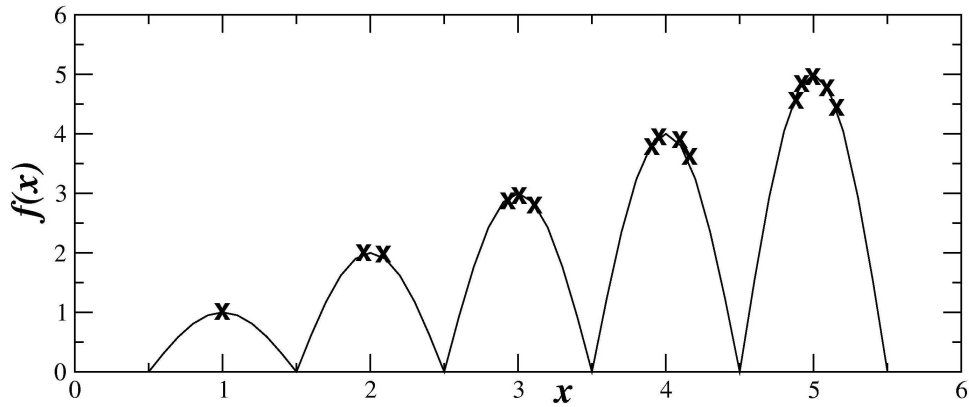


Figura 2.1: Disposición ideal del fitness sharing

Se puede observar que los óptimos locales con mayor valor de la función objetivo disponen de más individuos, lo cual es lógico, ya que los individuos localizados en los óptimos locales más altos tendrán inicialmente un mayor valor de fitness, pero este se penalizará en mayor medida ya que se encuentran cerca de un mayor número de individuos.

El Algoritmo 3 muestra el funcionamiento de esta técnica. Se puede observar que los pasos son los mismos que en el algoritmo genético original, a excepción de la adición del método de penalización del fitness de la nueva generación.

Aunque el fitness sharing es una técnica que promueve eficazmente la exploración del espacio de soluciones evitando la convergencia a un único óptimo global/local, cuenta con una limitación importante: requiere conocimiento de la función a priori. Para definir un valor  $\sigma$  (radio de los nichos) adecuado al problema, es necesario saber inicialmente la distancia entre los óptimos locales de la función.

De cualquier forma, el método se ha implementado para comprender su funcionamiento, sus limitaciones y comparar los resultados obtenidos al aplicarlo a los diferentes problemas propuestos en la competición.

---

**Algoritmo 3:** Fitness sharing

---

**Input :**  $np$  - tamaño de la población  
 $pm$  - probabilidad de mutación  
 $pc$  - probabilidad de cruce  
 $ss$  - método de selección de supervivientes  $mEval$  - número máximo de evaluaciones

```

1 P = ∅
2 Inicialización uniforme de P con np individuos
3 while mEval > eval do
4   Off = ∅
5   Off ← P.cruzar(pc);
6   Off = Off.mutar(pm);
7   Off.evaluar();
8   Off.penalizar();
9   P = selecciónSupervivientes(ss);
10 end

```

---

### 2.3.2. Crowding

El principal objetivo del crowding es el de mantener la diversidad de individuos entre generaciones, haciendo que, en cada generación, los individuos de la descendencia reemplacen aquellos individuos de la población a los que sean más similares. En el algoritmo original se genera un número de descendientes equivalente al 20 % del tamaño de la población. Cada nuevo individuo se compara con un subconjunto de individuos de la población y es intercambiado por aquel que sea más similar (el más cercano en el espacio de soluciones). La disposición ideal de individuos generada por el algoritmo es la mostrada en la Figura 2.2.

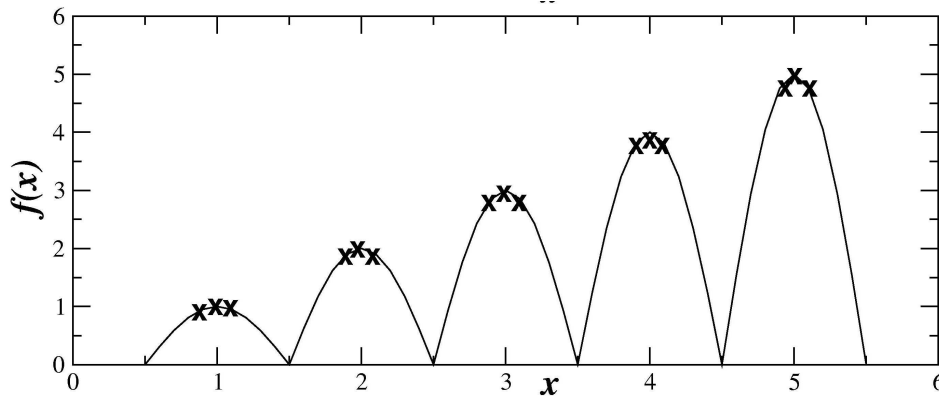


Figura 2.2: Disposición ideal del crowding

A diferencia del fitness sharing, los individuos se reparten de forma equitativa en los diferentes óptimos. Sin embargo, es difícil llegar a la disposición ideal ya que individuos muy buenos pueden ser intercambiados por otros mucho peores. Se han propuesto muchas alternativas y mejoras para el Crowding original. Una de ellas es el Deterministic Crowding (DC). El DC mantiene el principal concepto del Crowding pero propone nuevas ideas que mejoran considerablemente los resultados:

1. La población se empareja de forma aleatoria.
2. Cada pareja genera dos descendientes, a los que se les realiza la mutación y la evaluación.
3. Se calculan las distancias entre los padres ( $p_1$  y  $p_2$ ) y sus hijos directos ( $o_1$  y  $o_2$ ).
4. Se realiza un torneo binario entre los pares padre e hijo que minimicen la suma de las distancias calculada en la etapa anterior:

$$\begin{aligned} &bt(p_1, o_1) \ \& \ bt(p_2, o_2), \quad \text{si } (d(p_1, o_1) + d(p_2, o_2)) < (d(p_1, o_2) + d(p_2, o_1)) \\ &bt(p_1, o_2) \ \& \ bt(p_2, o_1), \quad \text{otherwise} \end{aligned}$$

Siendo  $d(p_1, o_1)$  la distancia entre  $p_1$  y  $o_1$  y  $bt(p_1, o_1)$  un torneo binario entre ellos.

De esta manera, la diversidad que proporciona el Crowding original se mantiene, y, al realizar torneos entre padres e hijos, se evita perder individuos que tengan muy buen fitness. Dado que en la literatura esta técnica obtiene resultados considerablemente mejores, se ha optado por implementar el DC en lugar del Crowding. El Pseudocódigo 4 muestra el funcionamiento del DC.

Finalmente, cabe mencionar que esta técnica ignora el parámetro de selección de supervivientes, ya que cada padre compete con uno de sus hijos directos para decidir quién sobrevive.

### 2.3.3. Crowding con clusters

Este algoritmo constituye la técnica que debía proponer el alumno para este TFG. El algoritmo implementado es muy similar al ‘Crowding clustering genetic algorithm’ propuesto en [6]. Esta técnica mezcla el concepto del Crowding de sustituir los individuos de la descendencia por los padres más cercanos con el uso de clusters explícitos con los que se puede interactuar. La técnica propuesta en [6] consta de los siguientes pasos:

**Algoritmo 4:** Crowding Determinístico

---

```

Input :    $np$  - tamaño de la población
            $pm$  - probabilidad de mutación
            $pc$  - probabilidad de cruce
            $mEval$  - número máximo de evaluaciones

1 P =  $\emptyset$ 
2 Inicialización uniforme de P con  $np$  individuos
3 while  $mEval > eval$  do
4   Off =  $\emptyset$ ;
5   Par =  $\emptyset$ ;
6   Par  $\leftarrow$  P.emparejarAleatoriamente();
7   for pareja en Par do
8     hijos =  $\emptyset$ 
9     hijos  $\leftarrow$  pareja.cruzar( $pc$ );
10    hijos = hijos.mutar( $pm$ );
11    Off  $\leftarrow$  torneoBinario(hijos, pareja);
12  end
13  P = Off;
14 end

```

---

1. La población de tamaño  $np$  se recombina y se muta para generar  $np$  nuevos individuos.
2. Se crea un vector de clusters (C), con un cluster por cada padre. Se incluye a cada hijo en el cluster del padre más cercano. En cada uno, el individuo con mayor fitness se selecciona como centro (CC), y el radio se calcula como la distancia del centro al individuo más alejado (CR).
3. Los clusters se ordenan dando prioridad a los individuos con mejor valor de fitness. En este caso, sabiendo que los problemas a resolver son de maximización, se ordenan de forma descendiente.
4. Se crea un segundo vector de clusters reservados (RC). Cada uno de los clusters de  $C_j$  es comparado con los reservados  $R_i$ . Si en todos los casos, el centro del cluster no se encuentra en el espacio de un cluster reservado ( $dist(CC_j, RCC_i) > RCR_i$ ) o el punto medio de ambos centros tiene mejor valor de fitness que la media de los valores objetivos de los extremos ( $\frac{f(CC_j+RCC_i)}{2} > \frac{f(CC_j)+f(RCC_i)}{2}$ ), este es incluido en el vector reservado. Además, el radio del nuevo cluster reservado se redefine como  $min(CR_j, dist(CC_j, RCR_i))$ .
5. Se crea una nueva población con los  $N_{RC}$  centros de los clusters reservados y  $Pob - N_{RC}$  nuevos individuos que deben crearse con una distribución uniforme por el espacio.

El algoritmo se ha implementado prácticamente como el propuesto en el artículo a excepción del modo del creación de clusters. El algoritmo original incluye a cada hijo en el cluster del padre más cercano. Esto implica que es necesario calcular la distancia de cada hijo hasta cada padre para averiguar cuál es la menor y asignarlo, así, al correspondiente. Si se utilizan poblaciones de muchos individuos es posible que este cálculo ralentice mucho al algoritmo, por lo tanto, se ha decidido utilizar la estrategia usada en el Crowding Determinista. Los padres se emparejan de forma aleatoria, y cada par genera dos descendientes. Cada descendiente será asignado al cluster de su padre directo más cercano. De esta manera, independientemente del tamaño de población, sólo se deberán calcular dos distancias por cada hijo. Su funcionamiento se explica en el Pseudocódigo 5.

Al utilizar esta estrategia de creación de clusters, es probable que se introduzcan individuos en clusters de padres que no sean el más cercano, por lo que los radios de estos aumentarán de forma general y es posible que en la etapa de filtrado de clusters reservados se eliminen individuos que en el algoritmo original se mantendrían. En la experimentación se estudiará si este hecho afecta demasiado negativamente a los resultados. Por otro lado el equilibrio entre exploración y explotación teóricamente se mantiene. La creación de nuevos individuos al final de cada generación fomenta la exploración de



zonas inexploradas, mientras que la selección de clusters debería ser capaz de mantener los individuos bien adaptados.

---

**Algoritmo 5: Clustering Crowding**

---

**Input** :  $np$  - tamaño de la población  
 $pm$  - probabilidad de mutación  
 $pc$  - probabilidad de cruce  
 $mEval$  - número máximo de evaluaciones

```
1 P =  $\emptyset$ 
2 Inicialización uniforme de P con np individuos
3 while  $mEval > eval$  do
4   Off =  $\emptyset$ ;
   /* Reproducción y creación de clusters */
5   Par =  $\emptyset$ ;
6   Par  $\leftarrow$  P.emparejarAleatoriamente();
7   C  $\leftarrow$  P.crearClusters();
8   for pareja en Par do
9     hijos =  $\emptyset$ 
10    hijos  $\leftarrow$  pareja.cruzar(pc);
11    hijos = hijos.mutar(pm);
12    añadirACluster(hijos, pareja);
13  end
14  C.redefinirCentros();
15  C.ordenar();
   /* Selección de mejores clusters */
16  RC =  $\emptyset$ ;
17  for c en C do
18    condicion = true;
19    for rc en RC do
20      if not( $d(c.centro, rc.centro) > rc.radio$  OR
21         $fitness(puntoMedio(c.centro, rc.centro) >$ 
22           $(fitness(c.centro) + fitness(rc.centro))/2)$ ) then
23        condicion = false;
24        break;
25      end
26    end
27    if (condicion) then
28      RC  $\leftarrow$  {c}
29    end
30  end
   /* Creación de la población de la nueva generación */
31  New =  $\emptyset$ ;
32  for  $i = 0 \rightarrow (np - RC.size())$  do
33    New  $\leftarrow$  crearIndividuoUniforme();
34  end
35  Off  $\leftarrow$  RC;
36  Off  $\leftarrow$  New;
37  P = Off;
end
```

---

## Capítulo 3

# Evaluación experimental

La experimentación se ha realizado según la metodología propuesta en la competición. Cada problema se ejecuta cincuenta veces con los mismos parámetros y se miden los ratios de acierto del algoritmo mediante tres métricas diferentes. La elaboración de las tablas de presentación de los resultados también se ha realizado basándose en las del enunciado de la competición<sup>1</sup>. Se recuerda que dichas métricas son las siguientes:

- **Convergencia (Conv):** Media de evaluaciones requeridas para localizar todos los óptimos globales. También se calcula la desviación estándar de los cincuenta valores obtenidos.
- **Peak Ratio (PR):** Proporción de óptimos globales encontrados con respecto al total.
- **Successful Runs (SR):** Proporción de ejecuciones exitosas (considerándose exitosa aquella ejecución en la que se localicen todos los óptimos globales).

Todos los experimentos se realizan exactamente con los mismos parámetros y se varía únicamente aquel parámetro que está siendo objeto de estudio. De esta manera, cualquier cambio en los resultados es, de forma inequívoca, ocasionado por el parámetro en cuestión. Los valores base de los parámetros usados en todos los experimentos son los siguientes:

- Tamaño de población: 200
- Probabilidad de mutación: 1
- Probabilidad de cruce: 1
- Selección de supervivientes generacional
- $\sigma$ : 1 (únicamente usado en el fitness sharing)

El máximo número de evaluaciones para cada problema también está definido por la competición. Para los problemas de  $F_1$  hasta  $F_5$ , se permiten un máximo de 50000 evaluaciones, para  $F_6$ ,  $F_7$ ,  $F_8$ ,  $F_9$  y  $F_{10}$  con dos dimensiones, se permiten 200000 y para el resto, 400000. Si en la tabla de convergencia aparece el número máximo de evaluaciones, se entiende que el algoritmo no ha logrado converger en ningún momento.

Inicialmente se pensaba hacer cada experimento con dos valores de  $\epsilon$  (precisión usada por la métrica para considerar que un individuo constituye óptimo global), pero en vista de que los resultados con ambos valores son bastante similares y que no iba a dar tiempo a realizarlos todos, se optó por hacer cada experimento una única vez. El valor de  $\epsilon$  usado en cada uno se especifica en las tablas.

### 3.1. Fitness sharing

Se han realizado experimentos variando los valores de tres parámetros diferentes:

---

<sup>1</sup><http://www.cec2017.org/>

- **Tamaño de población** tomando valores que varían entre 20 y 200 individuos (Tablas 6.1 y 6.2). Los valores entre 160 y 200 son los que mejores resultados obtienen, con hasta un 100 % de éxito en los problemas  $F_1$  y  $F_3$  y con convergencias que no superan las 600 iteraciones en las funciones  $F_1$ ,  $F_2$ ,  $F_3$  y  $F_4$ . En general parece que poblaciones grandes, con multitud de individuos funcionan mejor que poblaciones pequeñas, aunque el número de generaciones o iteraciones del algoritmo sea menor.
- **Radio del nicho** ( $\sigma$ ) tomando valores entre 0.3 y 15 (Tablas 6.3 y 6.4). Se ha elegido este rango de valores porque los espacios de búsqueda de las diferentes funciones son muy variables (oscilan entre 1 y 30 unidades). Este parámetro es muy dependiente de la función a la que se aplica el algoritmo, ya que, como se ha explicado, está relacionado con la distancia a la que se encuentran los óptimos entre sí, por lo que es complicado encontrar un valor que obtenga buenos resultados para todos los problemas. De cualquier forma, parece que los valores 0.3 y 0.6 tienen los mejores resultados en general, con un 90 % de ejecuciones exitosas en el segundo problema y un relativamente buen número de óptimos encontrados en las funciones  $F_7(2D)$ ,  $F_8(2D)$ ,  $F_9$ ,  $F_{10}$  y  $F_{11}(2D)$ .
- **Probabilidad de mutación** tomando valores entre 0 y 1 (Tablas 6.5 y 6.6). El valor 0.2 de este parámetro obtiene resultados bastante buenos. Entre los más destacables están un 50 % de ejecuciones satisfactorias en el problema  $F_{12}(3D)$ , con una convergencia de poco más de 40000 evaluaciones (casi diez veces menos de las máximas permitidas), así como un ratio general de óptimos encontrados considerablemente bueno en las funciones de dos, tres y cinco dimensiones. Los peores valores son 0 en algunos casos y 1 en otros, por lo que se recomienda usar valores intermedios.

Se ha observado un comportamiento interesante en las tablas. En todos los experimentos con los cinco primeros problemas siempre se logra converger en algún momento. Sabiendo esto, es de esperar que el algoritmo tenga éxito en todas estas ejecuciones. Sin embargo, en las tablas no aparece un 100 % de ejecuciones satisfactorias en todos los casos. En el siguiente capítulo se comenta detenidamente esta conducta.

En resumen, los problemas en los que este algoritmo ha logrado encontrar todos los óptimos son  $F_1$ ,  $F_2$ ,  $F_3$  y  $F_5$  con un 100 % de ejecuciones exitosas,  $F_{12}(3D)$  con un 50 % y  $F_4$ , con un 10 %.

### 3.2. Crowding determinista

En este caso sólo se han realizado los experimentos relativos al tamaño de la población y a la probabilidad de mutación, ya que el crowding no necesita parámetros de entrada propios.

- **Tamaño de la población** tomando valores entre 20 y 200 individuos (Tablas 6.7 y 6.8). De nuevo ocurre algo similar que en el fitness sharing, los valores 180 y 200 son los que más éxito tienen, consiguiendo el 100 % de aciertos en los problemas  $F_1$ ,  $F_2$  y  $F_3$  y los mejores ratios de óptimos encontrados en las funciones  $F_5$ ,  $F_7(2D)$  y  $F_8(2D)$ . Cabe destacar que en las tres primeras funciones, la convergencia se produce en aproximadamente la mitad de evaluaciones que con el fitness sharing. La conclusión a la que se llega es, de nuevo, que poblaciones más numerosas obtienen mejores resultados.
- **Probabilidad de mutación** tomando valores entre 0 y 1 (Tablas 6.9 y 6.10). Los mejores resultados se obtienen con los valores 0.8 y 1, que obtienen los mejores ratios de óptimos encontrados y ejecuciones exitosas en los problemas  $F_1$ ,  $F_2$ ,  $F_3$ ,  $F_5$ ,  $F_7(2D)$  y  $F_8(2D)$ , aunque las convergencias más rápidas no se logran con estos valores en todos los casos.

Se ha logrado un 100 % de ejecuciones exitosas en problemas  $F_1$ ,  $F_2$  y  $F_3$  y un 84 % en  $F_5$ . En el resto de problemas no se ha conseguido ninguna ejecución satisfactoria y los ratios de óptimos encontrados son bastante peores que los del fitness sharing.

### 3.3. Clustering crowding

Al igual que con el crowding determinístico, este algoritmo no tiene parámetros de entrada propios, por lo que sólo se realizaron dos experimentos:

- **Tamaño de la población** tomando valores entre 20 y 200 individuos (Tablas 6.11 y 6.12). Los resultados de las tablas no parecen verse tan afectados por este parámetro como en los algoritmos anteriores. Para todos los problemas a excepción del  $F_2$  (que es el único problema que parece verse claramente beneficiado de valores mayores de población) los ratios de óptimos encontrados no varían más de un 5% en todo el rango de valores del parámetro. También se observa que poblaciones de individuos más numerosas convergen más tarde, por lo que es posible que poblaciones grandes sólo entorpezcan al algoritmo.
- **Probabilidad de mutación** tomando valores entre 0 y 1 (Tablas 6.13 y 6.14). Este parámetro no parece afectar a los resultados de la misma forma para todos los problemas, parece el valor 0 es el acertado, haciendo que mejore de forma considerable el ratio de óptimos encontrados en los problemas  $F_5$ ,  $F_7(2D)$  y  $F_8(2D)$ . Además, este valor no afecta negativamente a las cuatro primeras funciones, ya que consiguen resultados muy similares en todos los casos.

Los mejores ratios de ejecuciones exitosas son un 100% en los problemas  $F_1$ ,  $F_2$ ,  $F_3$ , y  $F_4$ . En el resto de problemas no se ha conseguido ninguna ejecución satisfactoria.

### 3.4. Comparación de resultados

Para analizar los resultados de los algoritmos, compararlos entre sí y ver cuál tiene mayor índice de acierto, se ha creado la Tabla 3.1, en la que se muestran los mejores valores obtenidos por cada uno en cada problema. Los mejores resultados se han marcado en negrita.

Problema	Fitness sharing		Det. crowding		Clustering crowding	
	PR	SR	PR	SR	PR	SR
$F_1$	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>
$F_2$	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>
$F_3$	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>
$F_4$	0.33	0.10	0.04	0.00	<b>1.00</b>	<b>1.00</b>
$F_5$	<b>1.00</b>	<b>1.00</b>	0.92	0.84	0.72	0.44
$F_6(2D)$	0.05	0.00	0.00	0.00	<b>0.06</b>	<b>0.00</b>
$F_7(2D)$	0.03	0.00	<b>0.30</b>	<b>0.00</b>	0.20	0.00
$F_6(3D)$	<b>0.14</b>	<b>0.00</b>	0.00	0.00	0.01	0.00
$F_7(3D)$	0.00	0.00	0.01	0.00	<b>0.03</b>	<b>0.00</b>
$F_8(2D)$	<b>0.09</b>	<b>0.00</b>	0.05	0.00	0.06	0.00
$F_9(2D)$	<b>0.16</b>	<b>0.00</b>	0.00	0.00	0.00	0.00
$F_{10}(2D)$	<b>0.16</b>	<b>0.00</b>	0.00	0.00	0.00	0.00
$F_{11}(2D)$	<b>0.16</b>	<b>0.00</b>	0.00	0.00	0.00	0.00
$F_{11}(3D)$	<b>0.16</b>	<b>0.00</b>	0.00	0.00	0.00	0.00
$F_{12}(3D)$	<b>0.85</b>	<b>0.50</b>	0.00	0.00	0.00	0.00
$F_{11}(5D)$	<b>0.23</b>	<b>0.00</b>	0.00	0.00	0.00	0.00
$F_{12}(5D)$	<b>0.09</b>	<b>0.00</b>	0.00	0.00	0.00	0.00
$F_{11}(10D)$	<b>0.09</b>	<b>0.00</b>	0.00	0.00	0.00	0.00
$F_{12}(10D)$	0.00	0.00	0.00	0.00	0.00	0.00
$F_{12}(20D)$	0.00	0.00	0.00	0.00	0.00	0.00

Tabla 3.1: Comparación de mejores resultados de cada algoritmo

Se puede observar que los tres primeros problemas son resueltos perfectamente por los tres algoritmos, ya que son funciones sencillas. La cuarta función sólo es resuelta con éxito por el clustering crowding, superando, además, con creces los resultados del resto de algoritmos.

El fitness sharing consigue resolver la quinta función, imponiéndose esta vez sobre los demás algoritmos. En la  $F_{12}(3D)$  también tiene los mejores resultados con diferencia. En el resto de funciones

no se consigue ni una sola ejecución exitosa con ninguna técnica, pero en general el fitness sharing encuentra más porcentaje de óptimos, especialmente en las funciones  $F_9$ ,  $F_{10}$ ,  $F_{11}$ ,  $F_{12}$  en todas sus versiones.

## Capítulo 4

# Conclusiones y líneas futuras

Después de realizar la experimentación se ha comprobado que el fitness sharing ha sido, sorprendentemente, la técnica con mejores resultados en general, siendo el 50 % de ejecuciones exitosas en el problema  $F_{12}(3D)$ , el resultado más destacable, especialmente considerando que se trata de una función compleja e irregular con un total de ocho óptimos a encontrar.

El mayor problema encontrado en este algoritmo puede verse en la función  $F_4$ . En la Tabla 6.3, se observa que el algoritmo converge entre las 20000 y 25000 iteraciones en todos los casos, pero en la Tabla 6.4 aparecen ratios de acierto muy bajos. Esto es probablemente debido a que esta técnica no presenta una forma de mantener los óptimos encontrados durante las generaciones, de manera que converge a los óptimos de la función en un momento específico, pero los individuos no se mantienen en ellos durante las generaciones, y en la última iteración se han dispersado casi por completo.

Si en lugar de usar un enfoque generacional, se utiliza otra técnica de selección de supervivientes, como puede ser la elección de los individuos mejor adaptados entre padres e hijos (que ya está implementada en el algoritmo), es posible que se evite este problema, sin embargo, no se ha podido realizar esta experimentación por falta de tiempo. Un posible siguiente paso es el de experimentar con nuevas estrategias de selección de supervivientes, tanto en el fitness sharing como en el resto de algoritmos.

El crowding determinista ha sido el algoritmo que peores resultados ha dado en general. No destaca con respecto a los demás en ninguno de los problemas. Sabiendo esto, es posible que no resulte útil continuar investigando con él. Aún así, tal vez si se continuase con la experimentación se lograría encontrar algún valor de sus parámetros que mejorase los resultados, por lo que se podría intentar seguir por este camino.

Los resultados del clustering crowding han sido considerablemente malos. Se sabía que era una técnica arriesgada, ya que, como se ha comentado anteriormente, el hecho de no incluir cada hijo en el cluster de su padre más cercano hace que los radios de todos los clusters aumenten. Esto implica que, en el proceso de selección de supervivientes, es más probable que se desprecien individuos buenos. Aún así, quizás merezca la pena seguir experimentando con la técnica, considerando que es la única que consigue un 100 % de acierto en el problema  $F_4$  y que obtiene los mayores ratios de óptimos encontrados en el  $F_6$ .

Una posible mejora podría ser usar el concepto de selección de supervivientes del crowding original, y comparar cada hijo con un subconjunto aleatorio de la población, de esta manera se conservaría la idea que se perseguía con este algoritmo de evitar calcular la distancia desde cada hijo a cada padre y probablemente, los radios de los clusters disminuyan considerablemente. Es posible que este cambio sea suficiente para mejorar sus resultados y que pueda competir con los otros algoritmos. Otra posibilidad es la de seguir investigando cómo afecta el tamaño de la población al algoritmo e intentar sacar conclusiones a partir de ahí.

## Capítulo 5

# Conclusions and future lines of work

Experimental results have proved that, suprisingly, fitness sharing has been the best performing technique, with 50 % of successful runs being the most remarkable result, especially considering that it is a complex and irregular function with a total of eight global optima to find.

The major problem found in this algorithm can be seen in function  $F_4$ . Table 6.3 shows that the algorithm converges between 20000 and 25000 iterations in every case, however in Table 6.4 very low hit rates appear. This is probably due to the fact that this technique does not allow for a certain way of maintaining optima found over generations, so that it converges to the function optima at a specific moment, but individuals do not stay around them over generations, and have dispersed almost completely by the last generation.

If another survival selection technique is used instead of the generational approach, such as selecting the best individuals from parents and offspring (which is already implemented in the algorithm), this problem could probably be avoided. Nevertheless, these experiments have not been performed yet due to lack of time. The next step might be to experiment with new survival selection strategies, both with fitness sharing and the algorithms remaining.

Deterministic crowding has been the algorithm with the worst results overall. It does not stand out from the others in many of the problems. Knowing this, it would not be usefull to continue with research in this area. Even so, better results might be found by continuing the experimentation, so that could be a path to follow.

Clustering crowding result have been considerably unsuccessful. It was a risky technique to implement, since, as mentioned in previos chapters, not including every offspring in its closest parent's cluster, causes a general increase in the cluster's radius. which probably makes some good offspring be rejected in the survival selection process. Even so, it may be worthwhile to keep investigating with this technique, since it is the only one that achieves 100 % success in problem  $F_4$  and obtains the best optima found ratios in  $F_6$ .

One possible improvement could be to use the original crowding survivor selection concept, and compare each child to a random subset of the population This way the idea pursued with this algorithm to avoid calculating the distance from each child to each parent is preserved and probably, and the clusters' radius might diminish considerably. Possibly this change is enough to improve its results so that it can compete with the other algorithms.



## Capítulo 6

# Presupuesto

En abril de 2017 se definieron los objetivos y las tareas específicas que serían abarcadas en este TFG. Entre abril y junio de este mismo año se realizó una investigación general sobre algoritmos evolutivos y optimización multimodal y se implementó el fitness sharing. Estas tareas se realizaban de forma eventual, debido a la necesidad de atender a otras actividades docentes.

A partir de julio se comenzó a trabajar de forma más centrada en el proyecto dedicándole cinco y ocho horas al día. Durante este mes se comenzó con la experimentación del fitness sharing, y la redacción de los apartados de estado del arte de la memoria, se implementó el crowding determinístico y se investigaron técnicas de clustering para intentar proponer un nuevo algoritmo.

En agosto se implementó el crowding con clusters y se realizó la experimentación con este algoritmo y con el deterministic crowding, además de continuar con la elaboración de la memoria que no terminaría hasta principios de septiembre.

Durante este periodo, se calcula que se le han dedicado a la elaboración del TFG un total de 350 horas repartidas aproximadamente de la siguiente forma:

Horas dedicadas a las tareas		
Tarea	Tiempo dedicado (h)	Precio (euros)
<b>Investigación</b>		
Investigación general	50	400
Fitness sharing	20	160
Deterministic crowding	30	240
Crowding con clusters	40	320
<b>Implementación</b>		
Familiarización con el código	15	120
Fitness sharing	15	120
Deterministic crowding	20	160
Crowding con clusters	25	200
Métricas y scripts	15	120
<b>Experimentación y análisis de resultados</b>	70	560
<b>Redacción de la memoria</b>	50	400
<b>Total</b>	350	2800

Fitness Sharing: Convergencia con $\epsilon = 0.01$										
Pob.	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
20	8347.6	18190.06	50000.0	0.00	159.2	128.56	50000.0	0.00	13160.8	17620.56
40	6144.0	16195.02	41295.2	16215.04	154.4	111.43	50000.0	0.00	10960.0	19339.72
60	5173.2	14955.69	37164.0	20002.68	186.0	138.00	50000.0	0.00	8222.4	18251.09
80	3201.6	11823.60	11393.6	17951.59	209.6	143.90	50000.0	0.00	1329.6	6954.18
100	1248.0	6964.89	7888.0	16237.67	262.0	183.18	50000.0	0.00	1364.0	6949.50
120	1291.2	6964.45	5184.0	13339.45	268.8	204.44	50000.0	0.00	388.8	141.09
140	350.0	129.07	2783.2	9695.75	254.8	139.13	50000.0	0.00	428.4	135.47
160	358.4	68.33	617.6	548.70	281.6	190.82	50000.0	0.00	460.8	180.00
180	378.0	54.00	608.4	416.56	270.0	126.00	50000.0	0.00	504.0	209.91
200	432.0	73.32	596.0	400.98	284.0	155.38	50000.0	0.00	508.0	233.96
Pob.	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
20	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
40	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
60	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
80	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
100	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
120	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
140	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
160	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
180	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
200	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
Pob.	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
20	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
40	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
60	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
80	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
100	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
120	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
140	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
160	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
180	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
Pob.	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
20	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
40	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
60	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
80	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
100	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
120	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
140	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
160	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
180	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
200	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00

Tabla 6.1: Fitness sharing. Convergencia según tamaño de población ( $\epsilon = 0.01$ )

Fitness Sharing: Peak Ratio & Successful Runs $\epsilon = 0.01$										
Pob.	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
20	0.69	0.46	0.22	0.00	0.76	0.76	0.00	0.00	0.22	0.00
40	0.82	0.64	0.31	0.00	0.86	0.86	0.01	0.00	0.39	0.14
60	0.92	0.84	0.30	0.00	0.88	0.88	0.25	0.00	0.37	0.10
80	0.92	0.84	0.41	0.00	0.96	0.96	0.25	0.00	0.47	0.20
100	0.98	0.96	0.53	0.02	1.00	1.00	0.26	0.00	0.60	0.38
120	0.98	0.96	0.54	0.00	1.00	1.00	0.25	0.00	0.65	0.38
140	0.99	0.98	0.56	0.02	1.00	1.00	0.25	0.00	0.73	0.52
160	1.00	1.00	0.64	0.08	1.00	1.00	0.25	0.00	0.74	0.54
180	1.00	1.00	0.68	0.08	1.00	1.00	0.25	0.00	0.78	0.64
200	1.00	1.00	0.68	0.10	1.00	1.00	0.27	0.00	0.78	0.62
Pob.	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
20	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.08	0.00
40	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00
60	0.01	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.08	0.00
80	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00
100	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00
120	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00
140	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00
160	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00
180	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00
200	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00
Pob.	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
20	0.16	0.00	0.11	0.00	0.16	0.00	0.05	0.00	0.03	0.00
40	0.15	0.00	0.12	0.00	0.16	0.00	0.08	0.00	0.07	0.00
60	0.16	0.00	0.13	0.00	0.16	0.00	0.06	0.00	0.10	0.00
80	0.15	0.00	0.12	0.00	0.16	0.00	0.08	0.00	0.09	0.00
100	0.14	0.00	0.12	0.00	0.16	0.00	0.06	0.00	0.11	0.00
120	0.14	0.00	0.12	0.00	0.16	0.00	0.05	0.00	0.11	0.00
140	0.15	0.00	0.12	0.00	0.16	0.00	0.05	0.00	0.11	0.00
160	0.15	0.00	0.12	0.00	0.16	0.00	0.06	0.00	0.11	0.00
180	0.14	0.00	0.12	0.00	0.16	0.00	0.04	0.00	0.10	0.00
200	0.14	0.00	0.12	0.00	0.16	0.00	0.03	0.00	0.09	0.00
Pob.	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
120	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
140	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
160	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
180	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabla 6.2: Fitness sharing. PR y SR según tamaño de población ( $\epsilon = 0.01$ )

Fitness Sharing: Convergencia con $\epsilon = 0.01$										
$\sigma$	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0.3	210.0	70.00	520.0	97.98	550.0	150.00	25800.0	15620.14	292.0	127.81
0.6	500.0	0.00	570.0	223.83	550.0	150.00	21212.0	14560.13	296.0	134.10
1	500.0	0.00	550.0	180.28	550.0	150.00	22476.0	14881.88	252.0	87.73
2	500.0	0.00	560.0	162.48	550.0	150.00	23840.0	16173.21	272.0	111.43
3	500.0	0.00	560.0	162.48	510.0	70.00	24688.0	15114.66	260.0	100.00
4	500.0	0.00	550.0	150.00	540.0	135.65	23716.0	15469.67	280.0	126.49
5	500.0	0.00	520.0	97.98	580.0	183.30	20108.0	15190.11	268.0	130.29
7.5	500.0	0.00	560.0	162.48	520.0	97.98	22504.0	16605.02	288.0	139.48
10	500.0	0.00	610.0	270.00	530.0	118.74	22016.0	15064.49	268.0	110.34
12.5	500.0	0.00	550.0	150.00	530.0	118.74	25280.0	15987.95	264.0	101.51
15	500.0	0.00	530.0	118.74	520.0	97.98	23172.0	15543.08	256.0	106.13
$\sigma$	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0.3	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.6	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
1	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
2	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
3	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
4	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
5	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
7.5	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
12.5	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
15	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
$\sigma$	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0.3	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.6	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
1	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
2	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
3	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
4	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
5	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
7.5	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
10	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
12.5	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
15	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
$\sigma$	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0.3	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
0.6	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
1	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
2	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
3	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
4	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
5	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
7.5	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
10	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
12.5	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
15	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00

Tabla 6.3: Fitness sharing. Convergencia según  $\sigma$  ( $\epsilon = 0.01$ )

Fitness Sharing: Peak Ratio & Successful Runs con $\epsilon = 0.01$										
$\sigma$	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0.3	1.00	1.00	0.98	0.90	1.00	1.00	0.08	0.00	1.00	1.00
0.6	1.00	1.00	0.96	0.88	1.00	1.00	0.13	0.00	1.00	1.00
1	1.00	1.00	0.87	0.54	1.00	1.00	0.03	0.00	1.00	1.00
2	1.00	1.00	0.93	0.68	1.00	1.00	0.10	0.02	1.00	1.00
3	1.00	1.00	0.98	0.92	1.00	1.00	0.08	0.00	1.00	1.00
4	1.00	1.00	0.96	0.82	1.00	1.00	0.11	0.00	1.00	1.00
5	1.00	1.00	0.98	0.90	1.00	1.00	0.10	0.00	1.00	1.00
7.5	1.00	1.00	0.95	0.76	1.00	1.00	0.11	0.02	1.00	1.00
10	1.00	1.00	0.94	0.80	1.00	1.00	0.09	0.00	1.00	1.00
12.5	1.00	1.00	0.95	0.80	1.00	1.00	0.13	0.00	1.00	1.00
15	1.00	1.00	0.95	0.78	1.00	1.00	0.13	0.00	1.00	1.00
$\sigma$	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0.3	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.09	0.00
0.6	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.09	0.00
1	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.09	0.00
2	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.09	0.00
3	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.09	0.00
4	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.09	0.00
5	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.08	0.00
7.5	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.08	0.00
10	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.09	0.00
12.5	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.09	0.00
15	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.09	0.00
$\sigma$	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0.3	0.14	0.00	0.09	0.00	0.12	0.00	0.00	0.00	0.00	0.00
0.6	0.12	0.00	0.08	0.00	0.13	0.00	0.00	0.00	0.00	0.00
1	0.12	0.00	0.07	0.00	0.14	0.00	0.00	0.00	0.00	0.00
2	0.14	0.00	0.07	0.00	0.12	0.00	0.00	0.00	0.00	0.00
3	0.12	0.00	0.10	0.00	0.14	0.00	0.00	0.00	0.00	0.00
4	0.14	0.00	0.07	0.00	0.14	0.00	0.00	0.00	0.00	0.00
5	0.14	0.00	0.07	0.00	0.13	0.00	0.00	0.00	0.00	0.00
7.5	0.13	0.00	0.08	0.00	0.11	0.00	0.00	0.00	0.00	0.00
10	0.14	0.00	0.08	0.00	0.12	0.00	0.00	0.00	0.00	0.00
12.5	0.14	0.00	0.08	0.00	0.11	0.00	0.00	0.00	0.00	0.00
15	0.15	0.00	0.08	0.00	0.12	0.00	0.00	0.00	0.00	0.00
$\sigma$	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0.3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabla 6.4: Fitness sharing. PR y SR según  $\sigma$  ( $\epsilon = 0.01$ )

Fitness Sharing: Convergencia con $\epsilon = 0.01$										
$pm$	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0	452.0	96.42	2512.0	9698.59	200.0	0.00	1628.0	332.29	1296.0	6959.80
0.2	404.0	28.00	552.0	409.02	200.0	0.00	1804.0	497.18	300.0	175.50
0.4	448.0	129.98	1504.0	6934.00	200.0	0.00	2372.0	905.99	276.0	132.00
0.6	428.0	80.10	516.0	464.91	204.0	28.00	4012.0	1748.79	284.0	160.45
0.8	428.0	80.10	644.0	666.08	200.0	0.00	8572.0	6851.34	268.0	110.34
1	432.0	73.32	1588.0	6925.71	200.0	0.00	21844.0	16598.13	260.0	107.70
$pm$	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.2	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.4	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.6	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.8	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
1	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
$pm$	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	376300.0	93807.84
0.2	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	40088.0	106151.09
0.4	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.6	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.8	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
1	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
$pm$	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
0.2	384180.0	67715.92	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
0.4	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
0.6	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
0.8	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
1	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00

Tabla 6.5: Fitness sharing. Convergencia según probabilidad de mutación ( $\epsilon = 0.01$ )

Fitness Sharing: Probabilidad de mutación con $\epsilon = 0.01$										
$pm$	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0	1.00	1.00	1.00	1.00	1.00	1.00	0.33	0.04	0.98	0.96
0.2	1.00	1.00	1.00	1.00	1.00	1.00	0.30	0.10	1.00	1.00
0.4	1.00	1.00	1.00	1.00	1.00	1.00	0.10	0.00	1.00	1.00
0.6	1.00	1.00	1.00	1.00	1.00	1.00	0.10	0.00	1.00	1.00
0.8	1.00	1.00	1.00	1.00	1.00	1.00	0.09	0.00	1.00	1.00
1	1.00	1.00	1.00	1.00	1.00	1.00	0.13	0.00	1.00	1.00
$pm$	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0	0.05	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00
0.2	0.05	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00
0.4	0.05	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00
0.6	0.04	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00
0.8	0.02	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.09	0.00
1	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.09	0.00
$pm$	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0	0.16	0.00	0.10	0.00	0.15	0.00	0.05	0.00	0.04	0.00
0.2	0.16	0.00	0.12	0.00	0.16	0.00	0.16	0.00	0.85	0.50
0.4	0.16	0.00	0.12	0.00	0.16	0.00	0.16	0.00	0.16	0.00
0.6	0.16	0.00	0.12	0.00	0.16	0.00	0.16	0.00	0.13	0.00
0.8	0.16	0.00	0.12	0.00	0.16	0.00	0.03	0.00	0.12	0.00
1	0.16	0.00	0.08	0.00	0.11	0.00	0.00	0.00	0.11	0.00
$pm$	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.2	0.23	0.00	0.09	0.00	0.09	0.00	0.00	0.00	0.00	0.00
0.4	0.15	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.6	0.03	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabla 6.6: Fitness sharing. PR y SR según probabilidad de mutación ( $\epsilon = 0.01$ )

Crowding: Convergencia con $\epsilon = 0.1$										
Pob.	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
20	200.0	0.00	43008.4	17328.53	16040.8	23295.82	50000.0	0.00	47003.6	11860.09
40	200.0	0.00	15741.2	23163.19	7068.8	17321.59	50000.0	0.00	38019.2	21319.99
60	200.0	0.00	2077.2	9790.40	1068.0	6996.03	50000.0	0.00	35053.2	22892.72
80	200.0	0.00	92.8	29.33	83.2	15.68	50000.0	0.00	31044.8	24212.11
100	200.0	0.00	106.0	23.75	106.0	23.75	50000.0	0.00	19082.0	24205.15
120	200.0	0.00	124.8	23.52	124.8	23.52	50000.0	0.00	17116.8	23630.36
140	200.0	0.00	142.8	19.60	142.8	19.60	50000.0	0.00	13168.4	21903.10
160	200.0	0.00	163.2	22.40	163.2	22.40	50000.0	0.00	7203.2	17299.80
180	200.0	0.00	180.0	0.00	183.6	25.20	50000.0	0.00	7200.0	17284.95
200	200.0	0.00	200.0	0.00	200.0	0.00	50000.0	0.00	5216.0	14928.19
Pob.	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
20	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
40	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
60	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
80	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
100	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
120	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
140	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
160	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
180	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
200	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
Pob.	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
20	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
40	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
60	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
80	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
100	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
120	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
140	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
160	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
180	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
200	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
Pob.	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
20	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
40	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
60	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
80	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
100	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
120	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
140	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
160	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
180	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
200	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00

Tabla 6.7: Crowding Determinístico. Convergencia según tamaño de población ( $\epsilon = 0.1$ )



Crowding: SR y PR con $\epsilon = 0.1$										
<i>Pob.</i>	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
20	1.00	1.00	0.59	0.12	0.90	0.68	0.00	0.00	0.22	0.04
40	1.00	1.00	1.00	0.72	1.00	0.86	0.01	0.00	0.42	0.12
60	1.00	1.00	1.00	0.88	1.00	0.98	0.02	0.00	0.49	0.22
80	1.00	1.00	1.00	0.98	1.00	1.00	0.01	0.00	0.59	0.34
100	1.00	1.00	1.00	1.00	1.00	1.00	0.01	0.00	0.75	0.52
120	1.00	1.00	1.00	1.00	1.00	1.00	0.02	0.00	0.75	0.54
140	1.00	1.00	1.00	1.00	1.00	1.00	0.03	0.00	0.77	0.60
160	1.00	1.00	1.00	1.00	1.00	1.00	0.02	0.00	0.84	0.72
180	1.00	1.00	1.00	1.00	1.00	1.00	0.02	0.00	0.86	0.78
200	1.00	1.00	1.00	1.00	1.00	1.00	0.02	0.00	0.92	0.84
<i>Pob.</i>	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
20	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.01	0.00
40	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00
60	0.00	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00
80	0.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.02	0.00
100	0.00	0.00	0.17	0.00	0.00	0.00	0.00	0.00	0.03	0.00
120	0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.03	0.00
140	0.00	0.00	0.23	0.00	0.00	0.00	0.01	0.00	0.03	0.00
160	0.00	0.00	0.25	0.00	0.00	0.00	0.01	0.00	0.05	0.00
180	0.00	0.00	0.27	0.00	0.00	0.00	0.01	0.00	0.05	0.00
200	0.00	0.00	0.30	0.00	0.00	0.00	0.01	0.00	0.05	0.00
<i>Pob.</i>	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
120	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
140	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
160	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
180	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
<i>Pob.</i>	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
120	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
140	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
160	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
180	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabla 6.8: Crowding Determinístico. PR y SR según tamaño de población ( $\epsilon = 0.1$ )

Crowding: Convergencia con $\epsilon = 0.1$										
pm	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0	8336.0	18183.66	200.0	0.00	200.0	0.00	50000.0	0.00	10192.0	19904.13
0.2	7344.0	17210.56	200.0	0.00	200.0	0.00	50000.0	0.00	10188.0	19906.12
0.4	7344.0	17210.56	200.0	0.00	200.0	0.00	50000.0	0.00	8200.0	18243.16
0.6	7344.0	17210.56	200.0	0.00	204.0	28.00	50000.0	0.00	3248.0	11812.03
0.8	6352.0	16118.09	200.0	0.00	200.0	0.00	50000.0	0.00	4212.0	13502.32
1	8336.0	18183.66	200.0	0.00	200.0	0.00	50000.0	0.00	6208.0	16171.43
pm	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.2	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.4	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.6	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.8	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
1	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
pm	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.2	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.4	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.6	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.8	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
1	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
pm	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
0.2	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
0.4	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
0.6	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
0.8	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00
1	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00	400000.0	0.00

Tabla 6.9: Crowding Determinístico. Convergencia según probabilidad de mutación ( $\epsilon = 0.1$ )

Crowding: SR y PR con $\epsilon = 0.1$										
pm	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0	0.92	0.84	1.00	1.00	1.00	1.00	0.04	0.00	0.78	0.60
0.2	0.93	0.86	1.00	1.00	1.00	1.00	0.02	0.00	0.85	0.70
0.4	0.93	0.86	1.00	1.00	1.00	1.00	0.04	0.00	0.87	0.78
0.6	0.93	0.86	1.00	1.00	1.00	1.00	0.02	0.00	0.93	0.86
0.8	0.94	0.88	1.00	1.00	1.00	1.00	0.03	0.00	0.90	0.84
1	0.91	0.84	1.00	1.00	1.00	1.00	0.02	0.00	0.92	0.84
pm	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0	0.00	0.00	0.25	0.00	0.00	0.00	0.01	0.00	0.03	0.00
0.2	0.00	0.00	0.27	0.00	0.00	0.00	0.01	0.00	0.05	0.00
0.4	0.00	0.00	0.29	0.00	0.00	0.00	0.01	0.00	0.03	0.00
0.6	0.00	0.00	0.30	0.00	0.00	0.00	0.01	0.00	0.04	0.00
0.8	0.00	0.00	0.30	0.00	0.00	0.00	0.01	0.00	0.05	0.00
1	0.00	0.00	0.30	0.00	0.00	0.00	0.01	0.00	0.04	0.00
pm	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
pm	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabla 6.10: Crowding Determinístico. PR y SR según probabilidad de mutación ( $\epsilon = 0.1$ )

Clustering Crowding: Convergencia con $\epsilon = 0.1$										
Pob.	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
20	202.7	108.97	18150.9	23904.01	73.6	40.35	2056.1	775.39	34074.6	23248.56
40	246.4	99.59	174.6	97.90	127.1	12.83	3139.5	1052.78	10268.6	19892.62
60	364.8	152.94	130.9	92.67	91.6	56.32	3997.7	1518.27	10298.5	19889.37
80	400.1	176.24	128.6	89.45	93.8	46.96	5017.5	2145.63	2437.1	9730.88
100	442.8	146.10	133.3	82.73	104.6	32.34	5946.9	1665.32	1414.3	6964.18
120	498.8	182.62	130.9	53.51	120.0	0.00	7607.3	3121.72	488.6	338.18
140	558.3	216.02	147.0	49.28	153.7	67.04	7929.2	2685.57	376.1	403.97
160	617.5	177.49	160.0	0.00	168.2	57.54	8238.5	2600.09	408.2	448.07
180	657.0	169.77	180.0	0.00	180.0	0.00	8668.6	3142.52	1510.6	7008.95
200	715.3	176.94	200.0	0.00	200.0	0.00	9474.2	3164.70	413.1	479.65
Pob.	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
20	200000.0	0.00	200000.0	0.00	400000.0	00.00	400000.0	00.00	200000.0	0.00
40	200000.0	0.00	200000.0	0.00	400000.0	00.00	400000.0	00.00	200000.0	0.00
60	200000.0	0.00	200000.0	0.00	400000.0	00.00	400000.0	00.00	200000.0	0.00
80	200000.0	0.00	200000.0	0.00	400000.0	00.00	400000.0	00.00	200000.0	0.00
100	200000.0	0.00	200000.0	0.00	400000.0	00.00	400000.0	00.00	200000.0	0.00
120	200000.0	0.00	200000.0	0.00	400000.0	00.00	400000.0	00.00	200000.0	0.00
140	200000.0	0.00	200000.0	0.00	400000.0	00.00	400000.0	00.00	200000.0	0.00
160	200000.0	0.00	200000.0	0.00	400000.0	00.00	400000.0	00.00	200000.0	0.00
180	200000.0	0.00	200000.0	0.00	400000.0	00.00	400000.0	00.00	200000.0	0.00
200	200000.0	0.00	200000.0	0.00	400000.0	00.00	400000.0	00.00	200000.0	0.00
Pob.	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
20	200000.0	0.00	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00
40	200000.0	0.00	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00
60	200000.0	0.00	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00
80	200000.0	0.00	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00
100	200000.0	0.00	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00
120	200000.0	0.00	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00
140	200000.0	0.00	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00
160	200000.0	0.00	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00
180	200000.0	0.00	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00
200	200000.0	0.00	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00
Pob.	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
20	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00
40	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00
60	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00
80	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00
100	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00
120	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00
140	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00
160	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00
180	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00
200	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00	400000.0	00.00

Tabla 6.11: Crowding con clusters. Convergencia según tamaño de población ( $\epsilon = 0.1$ )

Clustering Crowding Peak ratio & Successful Runs $\epsilon = 0.1$										
Pob.	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
20	0.95	0.90	0.37	0.00	1.00	1.00	1.00	1.00	0.50	0.00
40	0.98	0.96	0.48	0.00	1.00	1.00	1.00	1.00	0.50	0.00
60	0.99	0.98	0.54	0.00	1.00	1.00	1.00	1.00	0.50	0.00
80	0.98	0.96	0.51	0.00	1.00	1.00	1.00	1.00	0.50	0.00
100	1.00	1.00	0.56	0.02	1.00	1.00	1.00	1.00	0.51	0.02
120	0.99	0.98	0.56	0.00	1.00	1.00	1.00	1.00	0.51	0.02
140	1.00	1.00	0.58	0.02	1.00	1.00	1.00	1.00	0.50	0.00
160	1.00	1.00	0.57	0.04	1.00	1.00	1.00	1.00	0.51	0.02
180	1.00	1.00	0.62	0.04	1.00	1.00	1.00	1.00	0.50	0.00
200	0.99	0.98	0.65	0.06	1.00	1.00	1.00	1.00	0.50	0.00
Pob.	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
20	0.06	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.06	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00
60	0.06	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.01	0.00
80	0.06	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.02	0.00
100	0.06	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.02	0.00
120	0.07	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.02	0.00
140	0.05	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.04	0.00
160	0.05	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.03	0.00
180	0.04	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.06	0.00
200	0.05	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.03	0.00
Pob.	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
120	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
140	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
160	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
180	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Pob.	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
120	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
140	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
160	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
180	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabla 6.12: Crowding con clusters. PR y SR según tamaño de población ( $\epsilon = 0.1$ )

Clustering Crowding: Convergencia con $\epsilon = 0.1$										
Pm	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0	735.3	209.26	200.0	0.00	200.0	0.00	6360.6	2058.84	575.7	612.31
0.2	724.4	241.78	200.0	0.00	200.0	0.00	6624.2	1917.42	415.4	312.62
0.4	706.3	191.44	200.0	0.00	200.0	0.00	7190.1	1999.07	429.6	397.59
0.6	689.4	160.30	200.0	0.00	200.0	0.00	7597.5	2117.52	463.0	427.72
0.8	705.2	172.39	200.0	0.00	200.0	0.00	8291.9	2845.58	411.6	338.79
1	705.5	172.26	200.0	0.00	200.0	0.00	10183.6	3248.48	309.4	263.91
Pm	$F_6(2D)$		$F_7(2D)$		$F_8(3D)$		$F_9(3D)$		$F_{10}(2D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.2	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.4	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.6	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
0.8	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
1	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00	200000.0	0.00
Pm	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.2	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.4	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.6	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.8	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
1	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
Pm	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D	Conv	St. D
0	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.2	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.4	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.6	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
0.8	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00
1	200000.0	0.00	200000.0	0.00	200000.0	0.00	400000.0	0.00	400000.0	0.00

Tabla 6.13: Crowding con clusters. Convergencia según probabilidad de mutación ( $\epsilon = 0.1$ )

Clustering Crowding: PR & SR $\epsilon = 0.1$										
P <sub>m</sub>	$F_1(1D)$		$F_2(1D)$		$F_3(1D)$		$F_4(2D)$		$F_5(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0	1.00	1.00	0.99	0.98	1.00	1.00	1.00	1.00	0.72	0.44
0.2	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.98	0.60	0.20
0.4	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.57	0.14
0.6	0.99	0.98	1.00	1.00	1.00	1.00	1.00	1.00	0.52	0.04
0.8	1.00	1.00	0.99	0.98	1.00	1.00	1.00	1.00	0.51	0.02
1	1.00	1.00	0.64	0.04	1.00	1.00	1.00	1.00	0.50	0.00
P <sub>m</sub>	$F_6(2D)$		$F_7(2D)$		$F_6(3D)$		$F_7(3D)$		$F_8(2D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0	0.05	0.00	0.20	0.00	0.01	0.00	0.02	0.00	0.06	0.00
0.2	0.05	0.00	0.19	0.00	0.01	0.00	0.03	0.00	0.05	0.00
0.4	0.05	0.00	0.17	0.00	0.01	0.00	0.02	0.00	0.04	0.00
0.6	0.05	0.00	0.14	0.00	0.01	0.00	0.01	0.00	0.05	0.00
0.8	0.09	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.04	0.00
1	0.06	0.00	0.06	0.00	0.00	0.00	0.01	0.00	0.05	0.00
P <sub>m</sub>	$F_9(2D)$		$F_{10}(2D)$		$F_{11}(2D)$		$F_{11}(3D)$		$F_{12}(3D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
P <sub>m</sub>	$F_{11}(5D)$		$F_{12}(5D)$		$F_{11}(10D)$		$F_{12}(10D)$		$F_{12}(20D)$	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabla 6.14: Crowding con clusters. PR y SR según tamaño de población ( $\epsilon = 0.1$ )

# Bibliografía

- [1] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. 2015.
- [2] C. Darwin. *The Origin Of Species*. November 1859.
- [3] Carlos Segura y Carlos I. González-Vila Eduardo Segredo, Ben Paechter. Optimization letters. 2017.
- [4] John H. Holland. *Adaptation in Natural and Artificial Systems*. 1992.
- [5] In X. Yao and D.B. Fogel. A. jain and d.b. fogel. case studies in applying fitness distributions in evolutionary algorithms. ii. comparing the improvements from crossover and gaussian mutation on simple neural networks. 2000.
- [6] Yang Zaiyue y Wang Qiuping Ling Qing, Wu Gang. Crowding clustering genetic algorithm for multimodal function optimization. 2006.
- [7] Hans-Georg BeyerHans-Paul Schwefel. *Evolution strategies – A comprehensive introduction*. 2002.
- [8] Robert E. Keller y Frank D. Francone Wolfgang Banzhaf, Peter Nordin. *Genetic Programming: An Introduction (The Morgan Kaufmann Series in Artificial Intelligence)*. 1998.
- [9] Michael G. Epitropakis Xiaodong Li, Andries Engelbrecht. Benchmark functions for cec'2013 special session and competition on niching methods for multimodal function optimization. 2016.
- [10] Rainer Storn y Kenneth Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. 1995.