

MÁSTER EN INVESTIGACIÓN EN INTELIGENCIA ARTIFICIAL

COMPUTACIÓN EVOLUTIVA

---

## Memoria Actividad 2

---

*Autora:*

Marta Rodríguez Sampayo



16 de febrero de 2020

# Índice

<b>1. Descripción del problema a resolver</b>	<b>3</b>
1.1. Esfera desplazada . . . . .	4
1.2. Función de Schwefel . . . . .	5
<b>2. Solución propuesta</b>	<b>5</b>
2.1. Algoritmos evolutivos . . . . .	6
<b>3. Resultados experimentales</b>	<b>8</b>
3.1. Función Esfera desplazada . . . . .	9
3.1.1. Parámetros iniciales . . . . .	9
3.1.2. Variaciones del tipo de mutación y selección de supervivientes . . . . .	9
3.1.3. Variación de los parámetros . . . . .	11
3.2. Función Schwefel . . . . .	13
3.2.1. Parámetros iniciales . . . . .	13
3.2.2. Variaciones en el tipo de mutación y selección de supervivientes . . . . .	13
3.2.3. Variación de los parámetros . . . . .	15
<b>4. Análisis y comparación de resultados</b>	<b>17</b>
4.1. Función esfera desplazada . . . . .	17
4.2. Función de Schwefel . . . . .	19
4.3. Análisis de la diferencia entre problemas . . . . .	20
<b>5. Conclusiones</b>	<b>20</b>
<b>6. Descripción del código</b>	<b>21</b>

# Índice de figuras

1. Representación de la función esfera (sin desplazamiento) . . . . .	4
2. Representación de la función de Schwefel . . . . .	5
3. Todas las combinaciones (2000 generaciones) . . . . .	9
4. Curvas de progreso (500 generaciones) . . . . .	10
5. Todas las combinaciones (250 generaciones) . . . . .	13
6. Curvas de progreso (500 generaciones) . . . . .	14

# Índice de tablas

1. Parámetros de configuración . . . . .	9
2. Mutación no correlacionada de paso único . . . . .	10
3. Mutación no correlacionada de N pasos . . . . .	10
4. Aumento del número de generaciones . . . . .	11
5. Aumento del tamaño de la población . . . . .	11
6. Variación de $\epsilon$ . . . . .	11
7. Variación de $\tau$ . . . . .	12
8. Variación de $\tau'$ . . . . .	12
9. Variación del tipo de recombinación . . . . .	12
10. Parámetros de configuración . . . . .	13
11. Mutación no correlacionada de paso único . . . . .	14
12. Mutación no correlacionada de N pasos . . . . .	14
13. Aumento del número de generaciones . . . . .	15
14. Aumento del tamaño de la población . . . . .	15

15.	Variación de $\epsilon$ . . . . .	15
16.	Variación de $\tau$ . . . . .	16
17.	Variación de $\tau'$ . . . . .	16
18.	Variación del tipo de recombinación . . . . .	16

## 1. Descripción del problema a resolver

En general, un problema de optimización consiste en la búsqueda de un conjunto de valores de entrada de una función real que permitan obtener su valor máximo o mínimo. Su definición generalizada es:

Dada una función  $f : A \rightarrow \mathbb{R}$  (siendo  $A$  un subconjunto de  $\mathbb{R}^n$ )

Se debe encontrar un elemento  $\mathbf{x}_0 \in A$  |  $f(\mathbf{x}_0) \leq f(\mathbf{x})$  o  $f(\mathbf{x}_0) \geq f(\mathbf{x}) \quad \forall \mathbf{x} \in A$

Además de pertenecer al espacio de búsqueda  $A$ , suelen especificarse una serie de restricciones que las posibles soluciones deben cumplir. La función  $f$  se denomina función objetivo, de pérdidas o de adaptación, entre otras, dependiendo del campo al que se aplique el problema. Finalmente, la solución que minimice (o maximice, dependiendo del objetivo) la función objetivo se denomina solución óptima y puede ser local, si minimiza la función en cierta región o global, si cumple estas condiciones para todo el dominio de la función. De aquí en adelante, se tratarán los problemas de optimización como problemas de minimización de funciones, puesto que las definiciones son análogas a los problemas de maximización.

Las funciones objetivo, pueden tener las siguientes características:

- Funciones convexas: si existe un mínimo local interior, este es un mínimo global. Mientras que en funciones no convexas, es posible que existan varios mínimos locales no necesariamente globales.
- Funciones multivariantes o multidimensionales: toman como entrada  $n$  números reales y calculan un único número real (valor de la función). Por tanto, se debe encontrar el valor de todas las variables de entrada que minimice la función.
- Funciones continuas: no poseen cambios abruptos (discontinuidades) entre los puntos de su dominio, es decir, su grafo es un conjunto conexo.
- Funciones monomodales o multimodales: solo existe un único mínimo local (y, por tanto, global) en el espacio de búsqueda. Al contrario, las funciones multimodales, poseen múltiples mínimos locales.

En esta memoria se presentan las soluciones desarrolladas para dos problemas de optimización. En ambos casos, el objetivo es encontrar el mínimo global de la función correspondiente. A continuación se definen dichas funciones objetivo.

## 1.1. Esfera desplazada

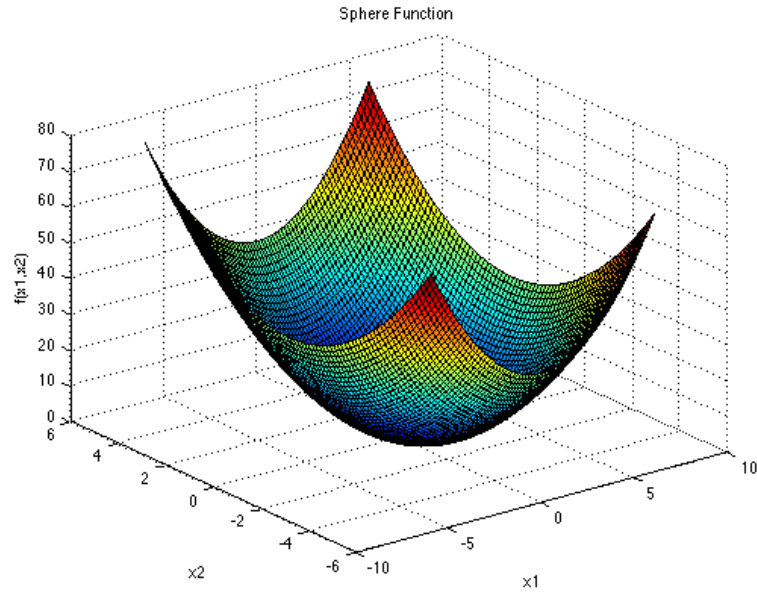


Figura 1: Representación de la función esfera (sin desplazamiento)

Cada función perteneciente a esta familia de funciones es multi-dimensional, continua, convexa y monomodal. Dado que esta función depende del desplazamiento aplicado, se utilizará la siguiente versión:

$$f(\bar{x}) = \sum_{i=1}^n (x_i - 10)^2$$

El dominio de definición se restringe al hipercubo  $-100 \leq x_i \leq 100, i = 1, \dots, n$ . El mínimo global corresponde al valor  $f(\bar{x}) = 0$  se alcanza en  $x_i = 10, i = 1, \dots, n$ .

En la figura 1 se representa la función esfera sin desplazamiento, pero la forma es idéntica para todas las funciones de esta familia, variando la situación del mínimo global (en este caso de 0 a 10) y el dominio de definición.

## 1.2. Función de Schwefel

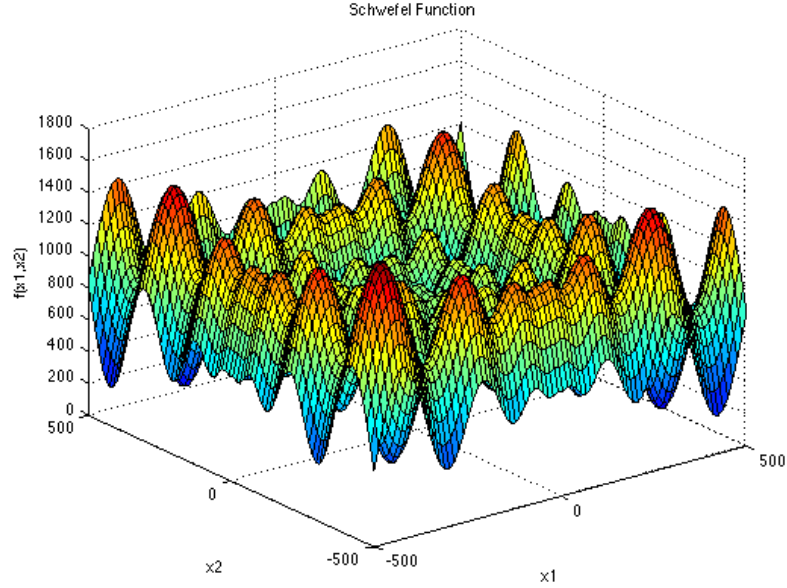


Figura 2: Representación de la función de Schwefel

Se trata de una función altamente multimodal, continua y no-convexa. Se define de la siguiente forma:

$$f(x) = 418,9829 \cdot n + \sum_{i=1}^n (-x_i \cdot \sin \sqrt{|x_i|})$$

donde  $|x_i|$  representa el valor absoluto de  $-500 \leq x_i \leq 500, i = 1, \dots, n$ . Tiene un mínimo global en  $x_i^* = 420,9687, i = 1, \dots, n$  y el valor de la función en este mínimo local es  $f(x_1^*, \dots, x_n^*) = 0$ .

En la figura 2 se puede apreciar la clara característica multimodal de la función de Schwefel.

## 2. Solución propuesta

A la hora de escoger un método para resolver problemas de optimización se deben tener en cuenta varios factores:

- Las características del problema: cada algoritmo se adapta a cierto tipo de problemas, pudiendo resultar más o menos eficaz.
- Eficiencia del algoritmo: dependiente de la información utilizada, la implementación concreta, el entorno de trabajo, etc.
- Precisión del algoritmo: hace alusión al nivel de ambigüedad en la secuencia de acciones del algoritmo.

Sin embargo, no existe una guía que seguir a la hora de escoger el más adecuado. A mayores, existen múltiples criterios para clasificar la gran variedad de algoritmos de optimización existentes [1], entre ellos cabe destacar la clasificación según la aleatoriedad empleada en la búsqueda de solución. De esta forma, es posible dividir estos métodos en dos grupos principales: algoritmos deterministas (algoritmo de escalada simple, método Nelder-Mead, etc.) o estocásticos (búsqueda lineal adaptativa, temple simulado, búsqueda aleatoria dinámica, búsqueda tabú, etc.). Entre estos últimos se encuentran los denominados

metaheurísticos, usualmente inspirados en la naturaleza y ampliamente utilizados, puesto que poseen numerosas ventajas frente a los algoritmos convencionales. En [2] se realiza un breve análisis sobre la elección de un algoritmo de optimización y se presentan brevemente varios algoritmos metaheurísticos; en lo que respecta a esta memoria, nos centraremos en un tipo particular, los algoritmos evolutivos.

## 2.1. Algoritmos evolutivos

Los algoritmos evolutivos (AEs) se basan en el principio de la selección natural, partiendo de una población inicial, esta "evoluciona" (se somete a ciertos cambios, prevaleciendo los individuos mejor adaptados) hasta alcanzar una solución óptima. Una introducción básica y breve a este tipo de algoritmos se muestra en [3], se puede encontrar información más detallada en [4]. En comparación a otros algoritmos de búsqueda de óptimos, los AEs se distinguen principalmente por el uso de una población, que define una función de distribución de probabilidad que dirige la generación de nuevos puntos en el espacio de soluciones (interacciones entre puntos existentes, variaciones de los mismos, etc.). Esto contrasta con las distribuciones uniformes globales o locales utilizadas por otros algoritmos estocásticos. Por ello, la capacidad de los AEs de mantener este conjunto diverso de puntos, permite evitar los óptimos locales y trabajar con espacios de búsqueda amplios y discontinuos. En conclusión, los algoritmos evolutivos proporcionan una forma natural y robusta de tratar problemas donde existe ruido o incertidumbre asociados a la asignación de una puntuación a una solución candidata [4].

Tal y cómo se ha expuesto en la sección 1, el método de resolución empleado debe facilitar la obtención de una serie de valores ( $n$ ) reales que minimicen la función objetivo y que se sitúen en el dominio de la misma. Por tanto, las variables con las que debe trabajar el sistema son vectores de números reales, de un tamaño fijo y dentro de unos límites estipulados. Dentro del campo de la computación evolutiva existen distintos tipos de algoritmos, adaptados a las diferentes características del problema. Los que mejor se adaptan a al supuesto planteado son las estrategias evolutivas [5].

A continuación se definen las particularidades de las EEs, siguiendo el esquema típico de los algoritmos evolutivos:

- Representación de los individuos de la población: vectores de  $n$  variables reales. Una característica especial de este tipo de algoritmos es la auto-adaptación de los parámetros de la estrategia, consistente en la variación de ciertos parámetros del algoritmo durante la ejecución. Con este fin, se incluyen en el genotipo de cada individuo los tamaños de paso de la mutación ( $\sigma$ ) y su número ( $n_\sigma$ , habitualmente 1 o  $n$ ) y los parámetros  $\alpha$  que representan las interacciones entre los diferentes  $\sigma$  (no se usan siempre).
- Recombinación: existen dos esquemas de recombinación según el número de padres utilizados.
  - Recombinación global: para cada componente  $i \in \{1, \dots, n\}$  de la descendencia se escogen aleatoriamente dos padres entre los  $\mu$  individuos de la población.
  - Recombinación local: cada hijo se genera a partir de dos padres seleccionados aleatoriamente.

El proceso de generación se repite hasta obtener  $\lambda$  descendientes y se puede llevar a cabo de dos formas. Dados dos vectores padre  $\mathbf{x}$  e  $\mathbf{y}$ , se obtiene  $\mathbf{z}$ :

- Recombinación discreta:  $z_i = x_i$  o  $y_i$
- Recombinación intermedia:  $z_i = \frac{x_i + y_i}{2}$

El esquema utilizado habitualmente es la recombinación global. Utilizando la variante discreta para las variables e intermedia para los parámetros de la estrategia. Este tipo es el utilizado en la solución implementada.

- Mutación: este es el operador de variación con mayor relevancia en EEs y está basado en la distribución normal  $N(0, \sigma)$ , de forma que los tamaños de paso de mutación coevolucionan junto a las soluciones y las variaciones pequeñas son más probables. Existen tres tipos básicos de mutación, se definen a continuación los dos utilizados en esta implementación:

- Mutación no correlacionada de paso único: solo un parámetro  $\sigma$  por cada individuo. El mecanismo de mutación se define según:

$$\sigma' = \sigma \cdot e^{\tau \cdot N(0,1)}$$

$$x'_i = x_i + \sigma' \cdot N_i(0,1)$$

donde  $\tau$  es una constante de proporcionalidad configurada por el usuario, cumpliendo

$$\tau \propto \frac{1}{\sqrt{n}}$$

Además, es recomendable que  $\sigma$  no sea muy próxima a cero, por lo que se suele establecer un umbral  $\epsilon$ .

- Mutación no correlacionada de N pasos: se utilizan diferentes tamaños de paso para diferentes dimensiones. Las fórmulas que definen este mecanismo son:

$$\sigma'_i = \sigma \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)}$$

$$x'_i = x_i + \sigma'_i \cdot N_i(0,1)$$

donde

$$\tau' \propto \frac{1}{\sqrt{2n}}$$

$$\tau \propto \frac{1}{\sqrt{2\sqrt{n}}}$$

De nuevo se aplica el umbral  $\epsilon$  para evitar desviaciones típicas muy cercanas a cero.

- Selección de padres: no se realiza en función del valor de adaptación de los individuos, si no de forma aleatoria siguiendo una distribución uniforme. Por tanto, los  $\mu$  individuos de la población son vistos como padres.
- Selección de supervivientes: después de generar la descendencia se seleccionan los  $\mu$  individuos con mejor valor de adaptación. Existen dos variantes, ambas utilizadas en el desarrollo de la solución:
  - $(\mu, \lambda)$ : la selección se realiza sobre el conjunto de descendientes.
  - $(\mu + \lambda)$ : la selección se realiza sobre la unión del conjunto de padres y descendientes.

El esquema usado generalmente es  $(\mu, \lambda)$ , las diferencias entre ambos se comentarán en la discusión de los resultados, en la sección 4 de esta memoria.

El valor de adaptación de cada individuo se calcula para cada uno de los problemas definidos en la sección 1, sustituyendo las  $n$  variables en la fórmula de la función correspondiente. Los valores de los parámetros utilizados se muestran en la tabla 1 para el algoritmo utilizado en la función de esfera desplazada y en la tabla 10 para la función de Schwefel.



### 3. Resultados experimentales

La evaluación de los experimentos realizados se realiza utilizando las siguientes métricas:

- *Success Rate (SR)*: porcentaje de ejecuciones terminadas con éxito.

$$SR = \frac{\text{Número de ejecuciones exitosas}}{\text{Número total de ejecuciones}} \cdot 100$$

- *Mean Best Fitness (MBF)*: media de los mejores valores de la función objetivo obtenidos al acabar cada ejecución.

$$MBF = \frac{1}{N} \sum_{i=1}^N BF_i$$

donde  $N$  es el número de ejecuciones y  $BF_i$  el mejor valor obtenido en cada ejecución.

- *Average number of Evaluations to a Solution (AES)*: el número medio de evaluaciones hasta una solución se calcula sobre el número de ejecuciones que han resultado en éxito, no sobre el total.

$$AES = \frac{\lambda}{N'} \sum_{i=1}^{N'} G_i$$

donde  $\lambda$  representa el número de descendientes generados (sobre los que se realizan las evaluaciones),  $N'$  el número de ejecuciones exitosas y  $G_i$  el número de generaciones necesarias para alcanzar la solución en cada ejecución.

- Tiempo de ejecución: la media (sobre todas las ejecuciones) del tiempo que tarda en ejecutarse una única vez el algoritmo completo (hasta finalizar el número de generaciones configurado), en segundos.
- Curvas de progreso: representan la variación del valor de adaptación (valor de la función objetivo) a medida que aumenta el número de generaciones. Para una mejor visualización de estas gráficas, en el eje Y (correspondiente a los valores de adaptación) se utiliza una escala logarítmica.

Tanto para el cálculo del SR como del AES, se debe definir qué se considera "éxito". Se definen los siguientes umbrales para cada una de las funciones:

- Función esfera desplazada: se considera que se ha alcanzado una solución si el valor de la función objetivo se encuentra en el intervalo  $[0, 0.0001]$ .
- Función de Schwefel: se considera que se ha alcanzado una solución si el valor de la función objetivo se encuentra en el intervalo  $[0, 0.01]$ .

Los resultados obtenidos para la función de esfera desplazada y la función de Schwefel se presentan respectivamente en los apartados 3.1 y 3.2. Cada uno de ellos se divide en 3 subsecciones, en la primera se muestran los valores de los parámetros con los que se inicializa el algoritmo, seguida de los resultados obtenidos variando el tipo de mutación y el método de selección de supervivientes y, finalmente, un apartado con los resultados obtenidos para diferentes valores de los parámetros del algoritmo.

### 3.1. Función Esfera desplazada

#### 3.1.1. Parámetros iniciales

Número de ejecuciones	20
Número de variables	10
Número de generaciones	2000
Tamaño de la población ( $\mu$ )	30
Tamaño de la descendencia ( $\lambda$ )	200
Límite inferior	-100
Límite superior	100
$\tau$	Paso único: $\frac{1}{\sqrt{10}}$ N pasos: $\frac{1}{\sqrt{2\sqrt{10}}}$
$\tau'$	$\frac{1}{\sqrt{20}}$
$\epsilon$	0.001
$\sigma$	1.0
Tipo de recombinación	Recombinación global (discreta + intermedia)

Tabla 1: Parámetros de configuración

#### 3.1.2. Variaciones del tipo de mutación y selección de supervivientes

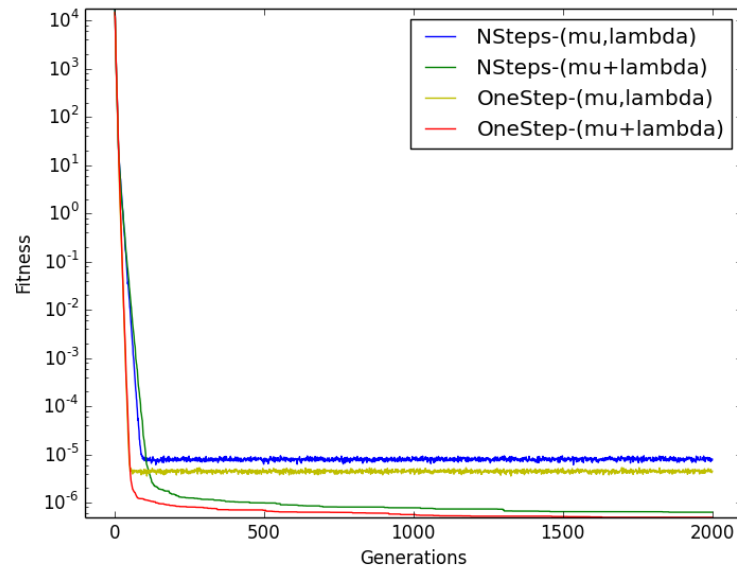
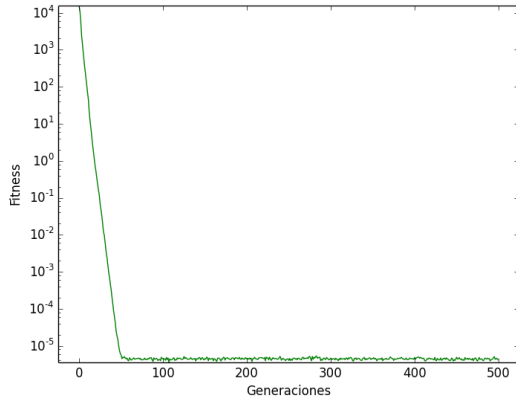
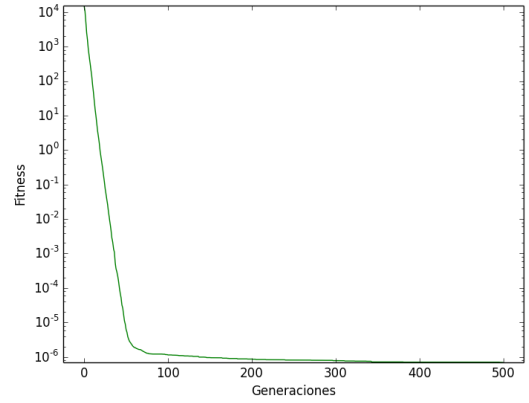


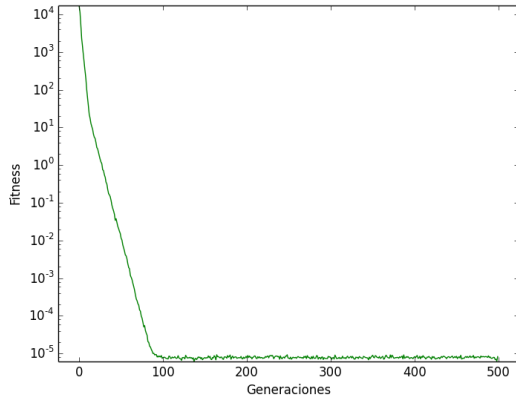
Figura 3: Todas las combinaciones (2000 generaciones)



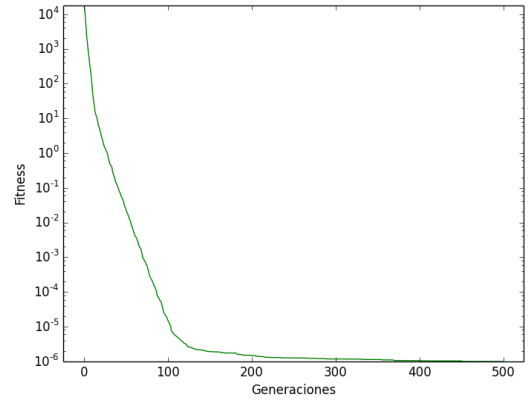
(a) Paso único -  $(\mu, \lambda)$



(b) Paso único -  $(\mu + \lambda)$



(c) N pasos -  $(\mu, \lambda)$



(d) N pasos -  $(\mu + \lambda)$

Figura 4: Curvas de progreso (500 generaciones)

SR	100 %
MBF	8.5604E-7
AES	7980
Tiempo de ejecución	1.5706

(a)  $(\mu, \lambda)$

SR	100 %
MBF	4.9340E-7
AES	8080
Tiempo de ejecución	1.6429

(b)  $(\mu + \lambda)$

Tabla 2: Mutación no correlacionada de paso único

SR	100 %
MBF	1.4928E-6
AES	14270
Tiempo de ejecución	2.0742

(a)  $(\mu, \lambda)$

SR	100 %
MBF	6.3213E-7
AES	16640
Tiempo de ejecución	2.2337

(b)  $(\mu + \lambda)$

Tabla 3: Mutación no correlacionada de N pasos

### 3.1.3. Variación de los parámetros

Esquema fijo escogido para realizar esta serie de experimentos:

- Mutación no correlacionada de paso único (salvo en la variación de  $\tau$  y  $\tau'$ )
- Selección de supervivientes ( $\mu, \lambda$ )

A continuación se presentan diferentes tablas con los resultados obtenidos tras ejecutar el algoritmo implementado, variando los parámetros especificados en cada una de ellas.

Número de generaciones	2000	4000	6000
SR	100 %	100 %	100 %
MBF	8.5603E-7	6.9031E-7	6.8020E-7
AES	7980	8060	8000
Tiempo de ejecución	1.5706	3.1511	4.7161

Tabla 4: Aumento del número de generaciones

Tamaño de la población	30	60	90	120	150
SR	100 %	100 %	100 %	100 %	100 %
MBF	8.5603E-7	1.0886E-6	1.5876E-6	2.4171E-6	5.8863E-6
AES	7980	22540	49140	107680	352850
Tiempo de ejecución	1.5706	1.9934	2.3912	2.6398	2.8876

Tabla 5: Aumento del tamaño de la población

$\epsilon$	0.1	0.05	0.01	0.002	0.001
SR	0	0	80 %	100 %	100 %
MBF	0.0455	0.01107	1.4568E-4	3.2965E-6	8.5603E-7
AES	0	0	114087.5	8170	7980
Tiempo de ejecución	1.6167	1.5957	1.6043	1.5856	1.5706

Tabla 6: Variación de  $\epsilon$

$\tau$	$\frac{1}{\sqrt{10}}$	$\frac{2}{\sqrt{10}}$	$\frac{3}{\sqrt{10}}$	$\frac{4}{\sqrt{10}}$	$\frac{5}{\sqrt{10}}$
SR	100 %	100 %	100 %	100 %	100 %
MBF	8.5603E-7	9.082E-7	8.6794E-7	8.9293E-7	9.5957E-7
AES	7980	6300	6040	6130	6200
Tiempo de ejecución	1.5706	1.6006	1.6333	1.5763	1.5895

(a) Variación de  $\tau$  (paso único)

$\tau$	$\frac{1}{\sqrt{2\sqrt{10}}}$	$\frac{2}{\sqrt{2\sqrt{10}}}$	$\frac{3}{\sqrt{2\sqrt{10}}}$	$\frac{4}{\sqrt{2\sqrt{10}}}$	$\frac{5}{\sqrt{2\sqrt{10}}}$
SR	100 %	100 %	100 %	100 %	0
MBF	1.4928E-6	3.8510E-6	1.1721E-5	4.9810E-5	0.00346
AES	14270	17070	22940	70400	0
Tiempo de ejecución	2.0742	2.0238	2.0601	2.2281	1.9552

(b) Variación de  $\tau$  (N pasos)

Tabla 7: Variación de  $\tau$

$\tau'$	$\frac{1}{\sqrt{20}}$	$\frac{2}{\sqrt{20}}$	$\frac{3}{\sqrt{20}}$	$\frac{4}{\sqrt{20}}$	$\frac{5}{\sqrt{20}}$
SR	100 %	100 %	100 %	100 %	100 %
MBF	1.4928E-6	1.2113E-6	1.1159E-6	9.3733E-7	9.6941E-7
AES	14270	8240	7120	6680	6830
Tiempo de ejecución	2.0742	1.9985	1.9955	2.1311	1.9787

Tabla 8: Variación de  $\tau'$

Tipo de recombinación	Global	Local
SR	100 %	100 %
MBF	8.5603E-7	8.5234E-7
AES	7980	9320
Tiempo de ejecución	1.5706	0.9623

Tabla 9: Variación del tipo de recombinación

## 3.2. Función Schwefel

### 3.2.1. Parámetros iniciales

Número de ejecuciones	20
Número de variables	10
Número de generaciones	2000
Tamaño de la población ( $\mu$ )	30
Tamaño de la descendencia ( $\lambda$ )	200
Límite inferior	-500
Límite superior	500
$\tau$	Paso único: $\frac{1}{\sqrt{10}}$
	N pasos: $\frac{1}{\sqrt{2\sqrt{10}}}$
$\tau'$	$\frac{1}{\sqrt{20}}$
$\epsilon$	0.001
$\sigma$	5.0
Tipo de recombinación	Recombinación global (discreta + intermedia)

Tabla 10: Parámetros de configuración

### 3.2.2. Variaciones en el tipo de mutación y selección de supervivientes

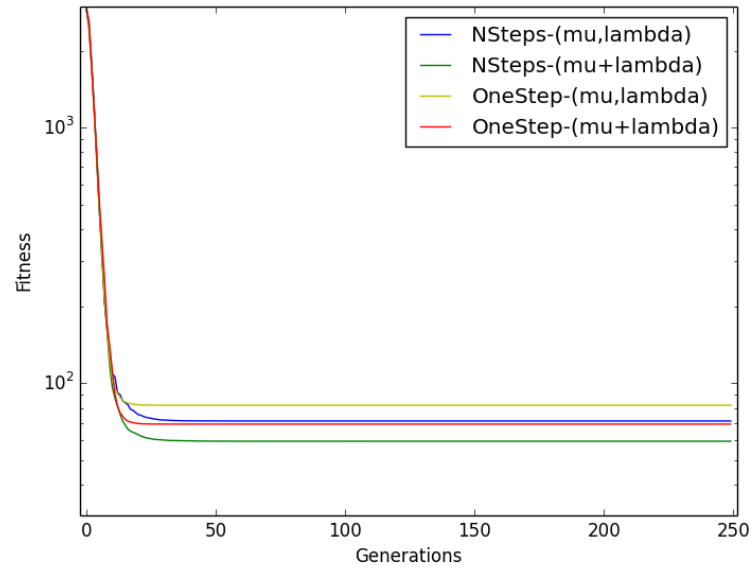
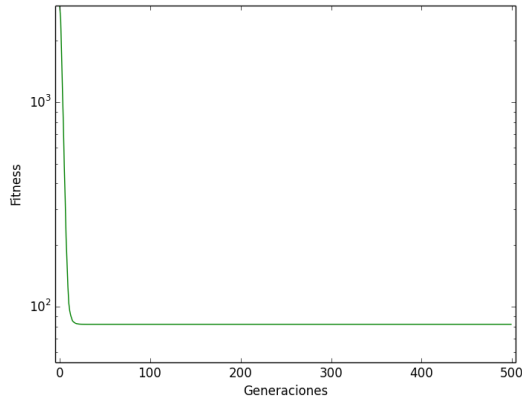
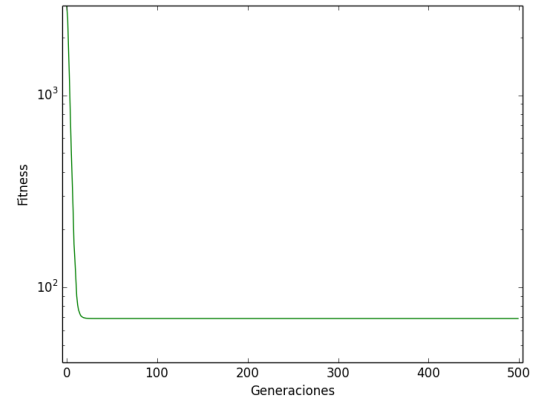


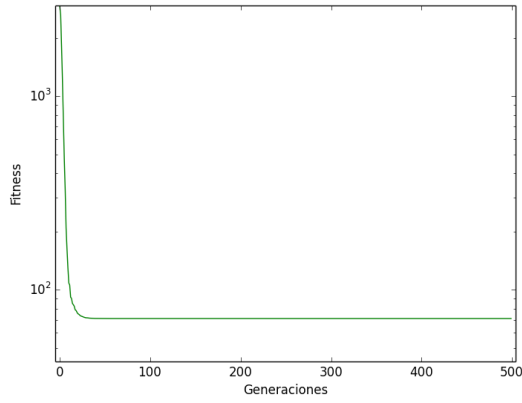
Figura 5: Todas las combinaciones (250 generaciones)



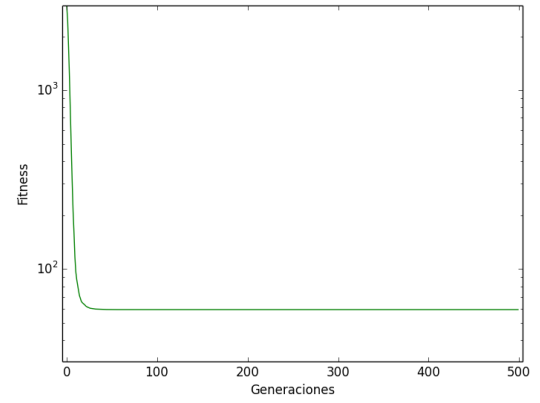
(a) Paso único -  $(\mu, \lambda)$



(b) Paso único -  $(\mu + \lambda)$



(c) N pasos -  $(\mu, \lambda)$



(d) N pasos -  $(\mu + \lambda)$

Figura 6: Curvas de progreso (500 generaciones)

SR	50 %
MBF	81.9201
AES	5680
Tiempo de ejecución	6.1462

(a)  $(\mu, \lambda)$

SR	65 %
MBF	69.0894
AES	5538.46
Tiempo de ejecución	6.9884

(b)  $(\mu + \lambda)$

Tabla 11: Mutación no correlacionada de paso único

SR	60 %
MBF	71.0631
AES	9566.66
Tiempo de ejecución	6.8845

(a)  $(\mu, \lambda)$

SR	55 %
MBF	59.2193
AES	10363.6
Tiempo de ejecución	7.6883

(b)  $(\mu + \lambda)$

Tabla 12: Mutación no correlacionada de N pasos

### 3.2.3. Variación de los parámetros

Esquema fijo escogido para realizar esta serie de experimentos:

- Mutación no correlacionada de paso único (salvo en la variación de  $\tau$  y  $\tau'$ )
- Selección de supervivientes ( $\mu, \lambda$ )

A continuación se presentan diferentes tablas con los resultados obtenidos tras ejecutar el algoritmo implementado, variando los parámetros especificados en cada una de ellas.

Número de generaciones	2000	4000	6000
SR	50 %	45 %	50 %
MBF	81.9201	111.5297	88.8289
AES	5680	5511.113	5520
Tiempo de ejecución	6.1462	12.3646	18.685

Tabla 13: Aumento del número de generaciones

Tamaño de la población	30	60	90	120
SR	50 %	95 %	95 %	95 %
MBF	81.9201	5.9220	5.9220	5.9220
AES	5520	15747.368	34768.4206	78989.47
Tiempo de ejecución	6.1462	10.3738	14.0638	16.7146

Tabla 14: Aumento del tamaño de la población

$\epsilon$	0.1	0.05	0.01	0.002	0.001
SR	40 %	45 %	50 %	50 %	50 %
MBF	82.9108	100.6735	106.5946	76.9850	81.9201
AES	5625	5644.446	5520	5560	5680
Tiempo de ejecución	6.1866	6.2416	7.2147	6.3354	6.1462

Tabla 15: Variación de  $\epsilon$



$\tau$	$\frac{1}{\sqrt{10}}$	$\frac{2}{\sqrt{10}}$	$\frac{3}{\sqrt{10}}$	$\frac{4}{\sqrt{10}}$	$\frac{5}{\sqrt{10}}$
SR	50 %	70 %	60 %	45 %	50 %
MBF	81.9201	76.9850	59.2193	76.9850	112.5165
AES	5680	4571.428	4416.66	4355.55	4640
Tiempo de ejecución	6.1462	6.34185	6.5452	6.2121	6.26115

(a) Variación de  $\tau$  (paso único)

$\tau$	$\frac{1}{\sqrt{2\sqrt{10}}}$	$\frac{2}{\sqrt{2\sqrt{10}}}$	$\frac{3}{\sqrt{2\sqrt{10}}}$	$\frac{4}{\sqrt{2\sqrt{10}}}$	$\frac{5}{\sqrt{2\sqrt{10}}}$
SR	60 %	55 %	35 %	0	0
MBF	71.0631	16.9850	224.3049	2107.6176	2536.05278
AES	1435	1810.9090	2485.7143	0	0
Tiempo de ejecución	6.8845	6.9872	6.7376	6.62085	6.6288

(b) Variación de  $\tau$  (N pasos)

Tabla 16: Variación de  $\tau$

$\tau'$	$\frac{1}{\sqrt{20}}$	$\frac{2}{\sqrt{20}}$	$\frac{3}{\sqrt{20}}$	$\frac{4}{\sqrt{20}}$	$\frac{5}{\sqrt{20}}$
SR	60 %	50 %	45 %	65 %	55 %
MBF	71.0631	87.8420	105.6078	41.4535	70.0763
AES	9566.66	5800	5133.33	5030.769	4945.454
Tiempo de ejecución	6.8845	7.076	6.7723	7.0027	6.87695

Tabla 17: Variación de  $\tau'$

Tipo de recombinación	Global	Local
SR	50 %	30 %
MBF	81.9201	161.8664
AES	5680	6433.33
Tiempo de ejecución	6.1462	5.5363

Tabla 18: Variación del tipo de recombinación

## 4. Análisis y comparación de resultados

Los valores de los parámetros iniciales de la estrategia evolutiva, mostrados en las tablas 1 y 10, se han escogido según los siguientes criterios:

- Número de ejecuciones, generaciones, número de variables,  $\mu$  y  $\lambda$ : propuestos en el enunciado de la actividad (más adelante se estudiará cómo afectan a los resultados).
- $\tau$  y  $\tau'$ : en el libro base de la asignatura [4] se especifica una relación de proporcionalidad para el valor de estos parámetros, por lo que se ha decidido utilizar, en una primera aproximación, la fórmula especificada para cada uno de ellos.
- $\epsilon$ : puesto que no se especifica ningún valor recomendado en la literatura consultada, se realizan varias ejecuciones con distintos valores (presentados en las tablas 6 y 15) y se selecciona el más adecuado. Debido a que los primeros experimentos se realizan sobre la función esfera, el valor de  $\epsilon$  escogido se utiliza también como línea base en la función de Schwefel.
- $\sigma$ : el tamaño inicial del paso de la mutación también se selecciona en base a resultados experimentales. Partiendo del valor 1, se aumenta una unidad hasta que la calidad de la solución alcanza un límite razonable.
- Tipo de recombinación: se escoge la recombinación global, utilizando recombinación discreta para las variables e intermedia para los parámetros. Este esquema es el utilizado habitualmente, ya que así se conserva la diversidad en el espacio de soluciones, permitiendo probar diferentes combinaciones de valores, mientras que el efecto de la recombinación intermedia asegura una adaptación más cautelosa de los parámetros de la estrategia [4].

En primer lugar se discutirán por separado los resultados obtenidos en cada uno de los problemas, para posteriormente comparar el comportamiento de la solución implementada en ambos.

### 4.1. Función esfera desplazada

En cuanto a la función de la esfera desplazada, se puede observar en la figura 3 que la convergencia del algoritmo es, en general, muy rápida, alcanzando valores muy cercanos a cero en las primeras generaciones y, en todos los casos, obteniendo una tasa de éxito del 100 %.

Las principales diferencias entre utilizar mutación de paso único o de N pasos, se aprecian tanto en las gráficas de la figura 4 como en los valores de las tablas 2 y 3. Por un lado, la pendiente de las curvas de progreso es más inclinada en el caso de la mutación de paso único, obteniendo valores más pequeños en un menor número de generaciones. Esto se ve reflejado en los valores de AES, considerablemente mayores para los casos de mutación de N pasos. Por otra parte, los valores de MBF obtenidos son mejores en el caso de un único paso (comparando las tablas que utilizan el mismo método de selección de supervivientes). En cuanto a los tiempos de ejecución, al tener que realizar la recombinación y mutación con un mayor número de parámetros  $\sigma$ , es lógico que aumente, disminuyendo la eficiencia del algoritmo. El método de mutación de N pasos, resulta más complejo, puesto que aumenta el número de parámetros que evolucionan con el algoritmo. Que este método no proporcione mejores resultados puede deberse a que los valores de las N  $\sigma$  tienden hacia un mismo valor, por lo que el método más sencillo (un único paso de mutación) cumple los requisitos para alcanzar las soluciones óptimas.

En relación al método de selección de supervivientes, los mejores resultados, en cuanto a MBF, se obtienen con  $(\mu + \lambda)$ . Los valores de AES aumentan, pero la diferencia es relativamente pequeña en comparación a los cambios entre los distintos tipos de mutación. El mismo efecto se observa en el tiempo de ejecución. La diferencia principal se visualiza en las curvas de progreso, por un lado tanto en la figura 4a como en la figura 4c se observan pequeños picos en el valor de la función objetivo, aunque siempre dentro de un margen razonable. Mientras que en las figuras 4b y 4d, la curva no muestra estas pequeñas oscilaciones, disminuyendo progresivamente con el aumento de número de generaciones. Este fenómeno se

debe a que utilizando el esquema  $(\mu + \lambda)$  el individuo mejor adaptado permanece en la población, siendo sustituido solo en el caso de que entre los descendientes se encuentre un individuo con mejor valor de la función de adaptación, por lo que los cambios en el *fitness* entre generaciones consecutivas son menos frecuentes, resultando en una curva más suave. El esquema utilizado habitualmente en EEs es  $(\mu, \lambda)$ , puesto que ofrece ventajas en el caso de topologías multimodales y funciones de adaptación que varíen con el tiempo [4]. Sin embargo, la función de esfera desplazada no contiene estas características, por lo que no es de extrañar que se obtengan buenos resultados con el esquema  $(\mu + \lambda)$ .

En conclusión, la configuración en la que se han obtenido mejores resultados experimentales se basa en la combinación del método de mutación no correlacionada de paso único y selección de supervivientes según el esquema  $(\mu + \lambda)$ . Se debe tener en cuenta que, al tratarse de valores tan pequeños, las diferencias son mínimas y pueden deberse a la aleatoriedad implícita en las estrategias evolutivas.

A mayores se han realizado diferentes experimentos variando los distintos parámetros de la EE, los resultados obtenidos se muestran en el apartado 3.1.3.

El incremento en el número de generaciones (tabla 4) y tamaño de la población (tabla 5) apenas afecta a los valores obtenidos de la función de adaptación. Como es evidente, los tiempos de ejecución sí aumentan, debido al mayor número de iteraciones o individuos que procesar. Se aprecia una leve mejoría en los valores de MBF al aumentar el número de generaciones, pero en balance con el aumento del tiempo de ejecución, el algoritmo se ve perjudicado. En el caso del aumento del tamaño de la población, el AES aumenta proporcionalmente, debido a que este parámetro se utiliza en su cálculo, pero no es relevante en cuanto a la evaluación de la eficacia del algoritmo.

Un parámetro que ha resultado ser decisivo en la precisión de las soluciones obtenidas es  $\epsilon$ , como se observa en la tabla 6. Aunque los valores de MBF son muy próximos a 0 (mínimo global), a medida que se disminuye el valor de este parámetro se acorta mucho más esta distancia, al mismo tiempo que decrece el valor del AES. Al observar estos resultados se ha realizado un experimento sin utilizar el parámetro  $\epsilon$  y, efectivamente, se alcanza el mínimo global (0 exacto) en pocas generaciones (aproximadamente 125). Al permitir valores de  $\sigma$  muy pequeños, las variaciones provocadas por la mutación son menores, permitiendo así precisar más el valor de la solución encontrada. Se debe tener en cuenta que el parámetro  $\epsilon$  aumenta la precisión del óptimo encontrado, ya sea global o local, por lo que resulta útil en funciones monomodales, como es este caso, pero no tendría por qué afectar en el caso de funciones multimodales. Este último caso se estudiará en la evaluación de los resultados de la función de Schwefel.

En cuanto al parámetro  $\tau$ , se ha estudiado cómo afecta al algoritmo utilizando los dos métodos de mutación descritos anteriormente y respetando, en cada caso, las relaciones de proporcionalidad que lo definen. En el caso de la mutación no correlacionada de paso único (tabla 7a), el aumento del valor de este parámetro no afecta al SR ni MBF y las diferencias entre los valores de AES son pequeñas, aunque se podrían achacar a una convergencia más rápida a medida que se aumenta el valor de  $\tau$ . Esta última observación tendría sentido puesto que en [4] se compara este parámetro con el factor de aprendizaje utilizado en redes neuronales, el cual produce el comportamiento indicado al aumentar su valor (mayor velocidad a costa de perder precisión en la solución).

En cambio, en el caso de la mutación de N pasos (tabla 7b), se reduce claramente la eficacia del algoritmo al aumentar  $\tau$ , disminuyendo el valor de MBF y aumentando el valor de AES. Sin embargo, el incremento del parámetro  $\tau'$  en este último escenario, disminuye el AES, es decir, se alcanza una solución en un menor número de generaciones, y permite obtener soluciones más próximas a 0. Este comportamiento se puede deber a que estos parámetros influyen en la evolución de los valores de los tamaños de paso de mutación. El incremento de  $\tau$  afecta a cada componente  $\sigma$  de forma diferente, puesto que para cada una se genera un número a partir de una distribución normal. Mientras que el parámetro  $\tau'$  produce un cambio constante en todas las  $\sigma$ .

Finalmente, en la tabla 9 se presentan los resultados obtenidos probando la recombinación global frente a la local. Los valores de SR, MBF y AES son prácticamente idénticos. En un problema de optimización

sencillo como este, la eficacia del algoritmo no se ve favorecida por la participación de más individuos en la generación de la descendencia.

## 4.2. Función de Schwefel

En las figuras 5 y 6 se observa de nuevo un rápido descenso en los valores de la función de adaptación durante las primeras generaciones que permanece constante a medida que aumentan. La tasa de éxito se sitúa alrededor del 50 %, manteniéndose el MBF en valores muy alejados del óptimo global.

Las diferencias entre los tipos de mutación no son muy notables. En general, se consiguen valores de MBF más pequeños empleando la mutación no correlacionada de  $N$  pasos, aunque también se aumentan el AES y el tiempo de ejecución (tablas 11 y 12), por los mismos motivos que ocurría en la función de la esfera. Al tratarse en este caso de un problema de optimización de mayor complejidad, la mutación de  $N$  pasos proporciona resultados mejores, puesto que permite un mayor grado de libertad a la hora de mutar los individuos de la población.

El esquema de selección de supervivientes  $(\mu + \lambda)$  parece obtener resultados más precisos, como se observa en las tablas 11b y 12b, el MBF es menor respecto al obtenido con el método  $(\mu, \lambda)$ . Los valores de SR en todos los casos son muy similares, pudiendo depender sus diferencias de la aleatoriedad del sistema, más que de la propia configuración del algoritmo.

De estos resultados se puede deducir que la mejor configuración para la estrategia evolutiva consiste en el uso de mutación no correlacionada de  $N$  pasos y el método de selección de supervivientes  $(\mu, \lambda)$ . Aunque el mínimo MBF se consigue con  $N$  pasos -  $(\mu + \lambda)$ , la tasa de éxito es menor y el AES mayor, por lo que se considera mejor estrategia la mencionada. Además, tomando como base la literatura consultada, esta configuración debería adaptarse mejor a los requisitos del problema.

Con el objetivo de afinar más la configuración de los parámetros de la EE, para la función de Schwefel también se han realizado diferentes experimentos variando sus valores.

Incrementar el número de generaciones no mejora los resultados obtenidos (tabla 13), puesto que el problema principal reside en la multimodalidad de la función, que se queda estancada alrededor de óptimos locales, no consiguiendo alcanzar el óptimo global en varias ejecuciones. Sin embargo, el aumento del tamaño de la población mejora notablemente la eficacia del algoritmo. En la tabla 14 se puede observar una mejora inmediata tanto en SR, pasando a un 95 % de ejecuciones exitosas, como en MBF mucho más cercano a cero. Este comportamiento puede ser debido al aumento del área del espacio de búsqueda que cubre el algoritmo, permitiendo explorar más soluciones posibles y evitando los mínimos locales. Sin embargo, esta mejora se aprecia al doblar el tamaño de la población, pero seguir aumentándolo no proporciona mejores resultados y aumenta considerablemente el tiempo de ejecución, disminuyendo la eficiencia.

Por otro lado, la variación del parámetro  $\epsilon$  no afecta al rendimiento de la implementación. Los valores de la tabla 15 son bastante lineales. Las diferencias de 5 % entre valores de SR no resultan muy significativas y lo mismo ocurre con AES y MBF.

Los resultados de las tablas 16 y 17 muestran un comportamiento muy similar. Aumentar los valores de estos parámetros favorece la velocidad de la convergencia, reduciendo los valores de AES. En cuanto a  $\tau$ , aumentar su valor afecta en mayor medida a la variante de  $N$  tamaños de paso de mutación, ya que al aumentarlo por encima de tres veces su valor inicial deja de alcanzarse el óptimo global, es decir, no se obtiene ningún éxito. Como se ha explicado anteriormente, este parámetro se considera el factor de aprendizaje del algoritmo, por lo que la precisión de las soluciones disminuye al aumentar la velocidad de convergencia, situándose los valores obtenidos entorno a óptimos locales. Los resultados obtenidos para  $\tau'$  no producen efectos relevantes, más allá de la velocidad de convergencia que se ha mencionado.

El último experimento se ha realizado alterando el tipo de recombinación empleado. La recombinación global vence claramente en esta comparativa. Los valores de las tres métricas empleadas, mostrados en la

tabla 18 son considerablemente mejores que para la recombinación local. La variante global permite una mayor diversidad entre la descendencia creada, puesto que es posible que un mayor número de padres participen en la recombinación.

### 4.3. Análisis de la diferencia entre problemas

Una vez realizado el análisis individual de cada una de las configuraciones de la estrategia evolutiva implementadas para cada uno de los problemas de optimización propuesto, resulta relevante comparar cómo afecta el tipo de función objetivo al comportamiento de la EE.

Es evidente que la función de esfera desplazada supone un desafío mucho menor a la hora de encontrar el óptimo global. La característica monomodal de esta función permite una convergencia rápida y con mayor precisión. Con la configuración más simple de una EE se alcanzan buenos resultados, no necesariamente mejorables variando las características principales (tipo de recombinación, mutación o selección de supervivientes) del algoritmo. En cuanto a los valores de los parámetros, el que más afecta a la implementación es sin duda  $\epsilon$ , que influye directamente sobre la precisión de las soluciones obtenidas. Eliminando este parámetro, de forma que se permita disminuir  $\sigma$  sin restricciones, se puede relajar el umbral de éxito a 0 exacto y obtener una tasa de éxito del 100 %.

En contraposición a esto se encuentra la función de Schwefel, altamente multimodal y, por tanto, presentando un problema mucho más complejo. En este caso, la decisión del tipo de estrategia evolutiva a seguir cobra mayor importancia. Ya no resulta tan relevante obtener soluciones precisas, es decir más o menos próximas a cero, si no que la convergencia se realice hacia el óptimo global y no hacia óptimos locales. Es por ello que se restringe el umbral de éxito, considerando buenas soluciones aquellas que se aproximen a 0.01, puesto que las dificultades para aproximarse a cero exacto son mayores. La clave para la implementación del algoritmo de resolución ha sido buscar una mayor exploración del espacio de búsqueda. Ejemplo de ello es la mejoría de los resultados al aumentar el tamaño de la población, utilizar mutación no correlacionada de  $N$  pasos o recombinación global. Al contrario que en la función de la esfera desplazada, donde una mayor explotación permite acercar más el valor obtenido al óptimo global.

## 5. Conclusiones

Los problemas de optimización han sido estudiados durante años, permitiendo el desarrollo de diferentes soluciones en campos muy variados. Como se ha expuesto en la sección 2, los metaheurísticos se han hecho un sitio entre estos métodos de resolución, ampliando las opciones posibles, más allá de solucionadores determinísticos. Puesto que no existe una fórmula exacta o una serie de reglas para decidir qué algoritmo utilizar para resolver un problema de optimización ni un método que resuelva con la misma calidad todos ellos, esta elección debe basarse en las características del problema y la evaluación experimental del método escogido.

Entre los distintos métodos existentes, se destacan los algoritmos evolutivos, objeto de esta memoria. La analogía con la evolución humana que representan favorece la diversidad entre soluciones posibles y el estudio más amplio del espacio de búsqueda. A mayores, las diferentes variantes que engloba esta familia de algoritmos permiten adecuar el método escogido al problema a tratar e, incluso, una vez escogido el tipo de algoritmo (algoritmo genético, estrategia evolutiva, programación evolutiva o programación genética), seleccionar sus características de forma que se aumente su rendimiento.

Por estos motivos se estudia el comportamiento de una estrategia evolutiva para tratar de minimizar dos funciones de varias variables reales. En la sección 2 se ha explicado por qué este método podría resultar adecuado para resolver los problemas planteados en el apartado 1, además de sus características principales. En este primer estudio se puede concluir que una EE cumple los requisitos necesarios para encontrar el óptimo global en ambos problemas. El análisis de los resultados de los diferentes experimentos que se realiza en la sección 4, demuestra que efectivamente, la implementación realizada de una EE permite obtener los valores que minimizan ambas funciones objetivos. Sin embargo, la sencillez del problema

de la función de la esfera desplazada, lleva a plantearse si utilizar este tipo de algoritmo conlleva un coste computacional demasiado elevado pudiendo escoger alternativas más simples que obtengan los mismos resultados consumiendo menos recursos. Por otra parte, la función de Schwefel presenta un problema más complejo, necesitando ajustar más la configuración de la estrategia evolutiva para conseguir buenos resultados. De esta forma, se puede afirmar que la solución implementada se adecúa al problema, solventando la complejidad del mismo y permitiendo una calidad razonable de las soluciones sin suponer un uso de recursos excesivo.

## 6. Descripción del código

La implementación de la estrategia evolutiva se ha realizado en el lenguaje de programación JAVA. La estructura de ficheros es la siguiente:

- Paquete *es (evolution strategy)*: aquí se incluye el código referente a las características propias de una estrategia evolutiva que se han empleado.
  - Clase *Mutation*: incluye dos métodos correspondientes a la implementación de los dos esquemas posibles de mutación no correlacionada, paso único y N pasos.
  - Clase *Recombination*: incluye dos métodos correspondientes a la implementación de los dos esquemas posibles de recombinación, global y local.
  - Clase *SurvivorSelection*: incluye dos métodos correspondientes a la implementación de los dos esquemas posibles de selección de supervivientes,  $(\mu, \lambda)$  y  $(\mu + \lambda)$ .
- Paquete *optimization-es*: engloba el código referente a los problemas de optimización planteados.
  - Clase *Main*: clase principal del programa. En ella se encuentran la definición de los valores de los parámetros del algoritmo y la secuencia de pasos a seguir para la ejecución y evaluación del mismo.
  - Clase *Menu*: implementación del menú que se muestra en el terminal al ejecutar el programa para seleccionar la función objetivo y la configuración de la estrategia evolutiva.
  - Clase *Evaluation*: cálculo de la tasa de éxito (SR), mejor valor de adaptación medio (MBF) y número medio de evaluaciones hasta una solución (AES).
  - Clase *FileManagement*: escritura de los archivos necesarios para evaluar los resultados posteriormente.
  - Clase *Individual*: objeto que define a un individuo de la población, incluyendo los valores de las variables, tamaños de paso de mutación ( $\sigma$ ) y valor de adaptación (calculado según la función objetivo seleccionada al inicio de la ejecución).

Al finalizar el número de ejecuciones especificado se crean los siguientes archivos dentro de la carpeta *Evaluation*:

- *aes.txt*: número de la generación en la que se alcanzó una solución. Una línea por cada ejecución exitosa.
- *evaluation.txt*: sigue el siguiente formato: SR MBF AES [valores de las variables y parámetros del individuo con mejor valor de adaptación] [nombre de la función objetivo] [ $\mu$ ] [ $\lambda$ ] [ $\sigma$  inicial] [tipo de mutación] [ $\tau$ ] [ $\tau'$ ] [ $\epsilon$ ] [esquema de selección de supervivientes].
- *progressCurvesX.txt*: en cada línea se escribe el número de generación y el mejor valor de adaptación obtenido en dicha generación. Se crea un archivo por cada ejecución.
- *times.txt*: tiempo de ejecución en milisegundos y mejor valor de adaptación conseguido en dicha ejecución.

Para procesar estos archivos se utilizan tres *scripts* en Python.

- *ProgressCurves.py*: a partir de los archivos *progressCurvesX.txt* se representa la curva de progreso del algoritmo (mejor valor de adaptación en cada generación). Se debe indicar el número de generaciones que se desea representar.
- *AverageTime.py*: calcula el promedio del tiempo de ejecución a partir del archivo *times.txt*.
- *ProgressCurvesOnePlot.py*: representa en una sola gráfica las curvas de progreso de las distintas variantes (cambiando mutación y selección de supervivientes) de una misma EE aplicada a una función objetivo.

Se ha utilizado este segundo lenguaje de programación porque ofrece un mayor número de opciones a la hora de representar gráficas y para separar el procesamiento de los resultados de la implementación del algoritmo.

Además del código y el ejecutable se incluye en la entrega una carpeta con el Javadoc generado, con información más detallada sobre la implementación.

Para llevar a cabo la ejecución del programa es necesario configurar los parámetros, el ejecutable incluido en la entrega tiene una configuración por defecto, para alterarla se debe ejecutar el programa con un argumento, por ejemplo un 0 y se mostrará un menú en la terminal donde se esté ejecutando para que el usuario pueda realizar las modificaciones que desee.

## Referencias

- [1] P. Pardalos and M. Resende, “Handbook of applied optimization,” 01 2002.
- [2] X.-S. Yang, *Optimization Algorithms*, 01 1970, vol. 356, pp. 13–31.
- [3] D. Soni, “Introduction to evolutionary algorithms,” 2018. [Online]. Available: <https://towardsdatascience.com/introduction-to-evolutionary-algorithms-a8594b484ac>
- [4] A. Eiben and J. Smith, *Introduction To Evolutionary Computing*, 01 2003, vol. 45.
- [5] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies - a comprehensive introduction,” *Natural Computing*, vol. 1, pp. 3–52, 03 2002.