

---

# Optimización Heurística

Evaluación de EAs

---

Daniel Tomás Sánchez  
Aarón Cabero Blanco  
Pablo Bautista Frías

08/01/2020

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Descripción del los algoritmos evolutivos</b>	<b>2</b>
<b>3. Distribución del código</b>	<b>3</b>
<b>4. Output de la ejecución</b>	<b>4</b>
4.1. Valores estadísticos . . . . .	4
4.1.1. Valores estadísticos DE . . . . .	4
4.1.2. Valores estadísticos SADE . . . . .	5
4.2. Evaluación de EAs . . . . .	6
4.2.1. Test de Kruskal . . . . .	6
4.2.2. Test de Mann Whitney-Wilcoxon . . . . .	7
4.3. Diagramas de cajas (boxplots) . . . . .	7
4.3.1. Diagrama de cajas - Sphere . . . . .	8
4.3.2. Diagrama de cajas - Ackley . . . . .	9
4.3.3. Diagrama de cajas - Rosenbrock . . . . .	10
4.3.4. Diagrama de cajas - Schwefel_2_22 . . . . .	11
<b>5. Conclusión</b>	<b>12</b>

# 1 | Introducción

En esta práctica, el objetivo es la comparación y evaluación de diferentes algoritmos evolutivos (EAs). Para ello, además de evaluarlos, se deberá previamente implementar un nuevo algoritmo evolutivo, el cual se explicará a continuación.

## 2 | Descripción del los algoritmos evolutivos

Por un lado, se nos pidió implementar nosotros a mano un nuevo algoritmo evolutivo. Respecto al algoritmo de la práctica anterior, el actual sigue siendo un DE (Differential Evolution) pero, en este caso, se nos pidió cambiar dos operadores: el operador de cruce y el operador de mutación. En este caso, el operador de cruce es binomial, frente al exponencial de la práctica 1, y el operador de mutación es de/best/1 frente al de/rand/1 de la anterior. Este algoritmo parece ser mejor que el de la práctica anterior ya que, a la hora de realizar la mutación, se emplea el mejor genoma de la población por lo que, a primera vista, parece que convergerá más rápido.

Por otro lado, a la hora de elegir un algoritmo de una biblioteca que nosotros quisiéramos, nos decantamos por el SADE debido a una gran fama en su eficiencia que, como veremos después, hemos podido comprobar nosotros mismos.

## 3 | Distribución del código

El código contiene toda la parte del algoritmo implementado por nosotros y un programa principal que es el que ejecutará las funciones de esta práctica. Este programa principal, denominado *benchmarking.py* es el encargado de realizar la funcionalidad pedida y que, tras ejecutarla, crea diferentes archivos:

- **benchmarking.out**: contiene los resultados de nuestro algoritmo y del SADE, en las secciones *BASIC\_DE* y *SADE* respectivamente, los valores estadísticos (media, mediana, min, max y desv\_est) de los dos anteriores, los resultados del test de kruskal y los fitness medios de cada función. Para consultar estos valores, como es muy extenso el archivo se recomienda abrir el archivo *benchmarking.out*.
- **tableDE.png**: es la tabla que contiene los valores estadísticos nombrados anteriormente para cada función cuando se usa optimiza mediante nuestro DE.
- **tableSADE.png**: es la tabla que contiene los valores estadísticos nombrados anteriormente para cada función cuando se usa optimiza mediante nuestro SADE.
- **tableWhitney.png**: es la tabla que contiene el test de Wilcoxon comparando ambos algoritmos.

Además, se generan diferentes diagramas de cajas comparando los dos algoritmos juntos por cada función.

## 4 | Output de la ejecución

En este apartado vamos a ver cuáles son las diferentes salidas que tiene nuestro código. Para ello, veremos las diferentes tablas que se generarán junto con una breve descripción explicando cada una de ellas. Además, se analizará algún diagrama de cajas para, como dijimos antes, comparar los estadísticos típicos de los algoritmos en cada función.

### 4.1 Valores estadísticos

Aquí veremos las tablas con los valores estadísticos tanto para nuestro algoritmo como para el SADE.

Hemos creado un código de color para los valores medios. Verde (los mejores) para valores menores que 1, naranja/amarillo (los buenos/aceptables) para los valores entre 1 y 10, negro (malos/normales) para los valores entre 10 y 100 y por último rojo (muy malos) para los valores mayores que 100.

#### 4.1.1 Valores estadísticos DE

	mean	median	min	max	desv_est
sphere	9.5856e+02	8.4646e+02	1.0030e+02	2.7582e+03	6.6624e+02
ackley	2.0012e+01	2.0012e+01	2.0001e+01	2.0029e+01	6.6908e-03
rosenbrock	3.5529e+07	3.1868e+07	1.2214e+06	9.6814e+07	2.5455e+07
rastrigin	1.5115e+03	1.3998e+03	5.2219e+02	2.7732e+03	6.5737e+02
griewank	1.2756e+00	1.2496e+00	1.0264e+00	1.8132e+00	1.8849e-01
schwefel_2_21	3.0033e+01	2.9146e+01	1.6438e+01	4.4925e+01	7.1305e+00
schwefel_2_22	1.8538e+04	1.5923e+02	1.8935e+01	4.0170e+05	8.0650e+04
schwefel_1_2	2.9970e+03	2.9088e+03	9.4673e+02	5.5669e+03	1.3875e+03
extended_f_10	4.5771e+01	4.6748e+01	2.6294e+01	6.2804e+01	9.9385e+00
bohachevsky	3.3519e+03	3.3694e+03	8.8009e+02	6.2403e+03	1.6358e+03
schaffer	3.8874e+01	3.8591e+01	2.2594e+01	5.4944e+01	8.4937e+00

En esta tabla se encuentran los valores estadísticos de cada función. En casi todas ellas, los valores *min* y *max* se diferencian mucho del valor medio (desv\_est alta) lo que nos dice que es heterogéneo. El mejor valor medio es

el de *griewank*, señalado en verde, pero esto no nos dice nada ya que son funciones diferentes que pueden converger de diferente forma. Por otro lado, la función más homogénea es la *ackley* que se verá posteriormente.

#### 4.1.2 Valores estadísticos SADE

	mean	median	min	max	desv_est
<b>sphere</b>	3.2276e-12	4.0107e-13	9.7990e-16	3.3553e-11	7.0263e-12
<b>ackley</b>	9.8084e-07	5.3061e-07	5.1349e-08	4.1802e-06	1.0898e-06
<b>rosenbrock</b>	3.5226e+00	3.6482e+00	5.1204e-02	6.9092e+00	1.8429e+00
<b>rastrigin</b>	5.6370e-01	9.4028e-02	3.7986e-07	4.3259e+00	1.0122e+00
<b>griewank</b>	1.0000e+00	1.0000e+00	1.0000e+00	1.0000e+00	5.2267e-15
<b>schwefel_2_21</b>	2.3701e-04	1.3116e-04	2.0984e-05	1.5779e-03	3.2224e-04
<b>schwefel_2_22</b>	5.1921e-07	3.7345e-07	5.6535e-08	1.3081e-06	4.2425e-07
<b>schwefel_1_2</b>	7.4720e-05	5.4645e-06	1.7783e-07	1.1223e-03	2.2588e-04
<b>extended_f_10</b>	6.5321e-02	5.7129e-02	2.8102e-02	1.1980e-01	2.5406e-02
<b>bohachevsky</b>	2.0546e-10	5.2106e-11	1.1124e-13	1.6040e-09	4.2592e-10
<b>schaffer</b>	6.0562e-02	5.3722e-02	2.5039e-02	1.2390e-01	2.4572e-02

Al igual que en la imagen anterior, se muestran los valores estadísticos pero, en este caso, para el SADE. Vemos que esta vez hay muchos valores medios cercanos al 0 lo que nos demuestra una mejor eficiencia del algoritmo. Por otro lado, la desv\_est es menor, cercana a 0 en casi todas las funciones que nos muestra una mayor homogeneidad de los resultados frente a los resultados obtenidos anteriormente con el DE.

## 4.2 Evaluación de EAs

Cabe destacar que en cuanto a la evaluación de EAs, únicamente se va a realizar el test de Kruskal y el de Wilcoxon, pero no el de Friedman ya que para él es necesario un tercer algoritmo con el que comparar, cosa que no tenemos.

### 4.2.1 Test de Kruskal

El test de Kruskal es un test no paramétrico para probar si un grupo de datos proviene de la misma población. Se asume, bajo la hipótesis nula, que los datos vienen de la misma distribución. A continuación se muestran los datos para nuestros algoritmos:

Results for the sphere function: 1.3328142940540802e-09  
Results for the ackley function: 1.3328142940540802e-09  
Results for the rosenbrock function: 1.3328142940540802e-09  
Results for the rastrigin function: 1.3328142940540802e-09  
Results for the griewank function: 7.019764390051665e-10  
Results for the schwefel\_2\_21 function: 1.3328142940540802e-09  
Results for the schwefel\_2\_22 function: 1.3328142940540802e-09  
Results for the schwefel\_1\_2 function: 1.3328142940540802e-09  
Results for the extended\_f\_10 function: 1.3328142940540802e-09  
Results for the bohachevsky function: 1.3328142940540802e-09  
Results for the schaffer function: 1.3328142940540802e-09

Como vemos, todos ellos tienen un p-valor muy cercano al 0 lo que nos indica que: uno, los test, efectivamente, se han realizado con la misma población de datos y, dos, que los fitness de las funciones tienen diferencias significativas por lo que no se puede asumir que el fitness en todas esas funciones sea el mismo. Por ello, se puede aplicar el Test de Mann-Whitney-Wilcoxon.

### 4.2.2 Test de Mann Whitney-Wilcoxon

El test de Mann Whitney-Wilcoxon con corrección de Holm, al igual que el de Kruskal, es un test no paramétrico que consiste en ver si hay diferencias significativas en el comportamiento de ambos algoritmos. Para ello, se toma la hipótesis nula de que la diferencia de las medianas de las dos distribuciones es 0.

	DE	SADE
DE	1.000000	0.000977
SADE	0.000977	1.000000

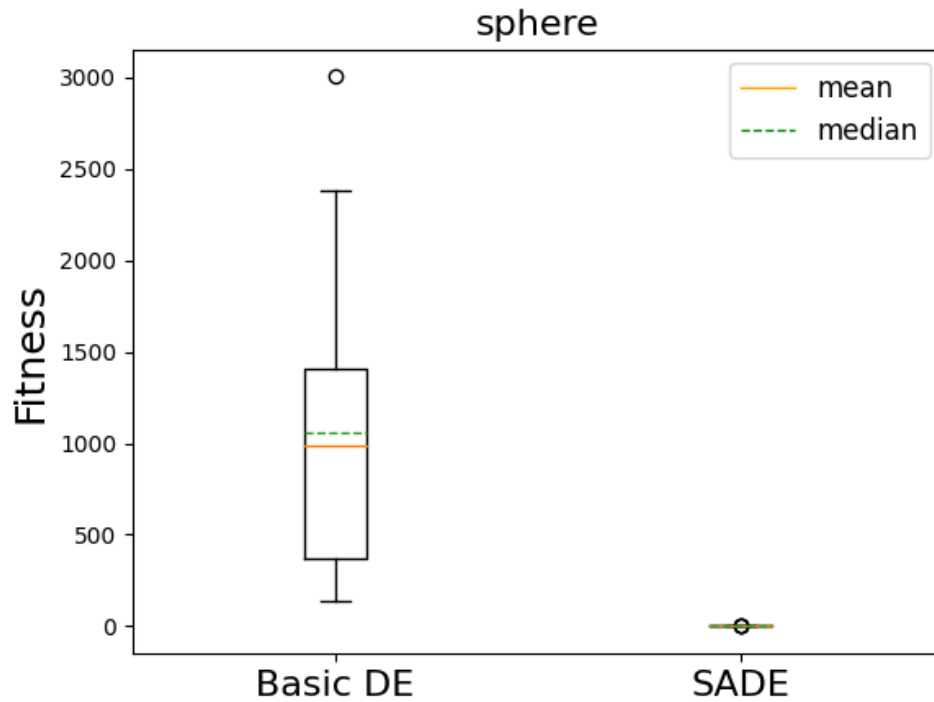
Como vemos en la tabla, el p-valor comparando ambos algoritmos es muy cercano a 0 y menor que el 0.05 % lo que nos ratifica que el SADE es sustancialmente mejor que nuestro DE.

## 4.3 Diagramas de cajas (boxplots)

A continuación vamos a comparar los diferentes algoritmos viendo los diagramas de cajas en diferentes funciones significativas. Para ello se realizarán mediante la ejecución de 25 repeticiones, cada una de ellas con 250 iteraciones. Este pequeño número de iteraciones sirve para ver la diferencia de eficiencia y convergencia entre ambos algoritmos ya que, con estas pocas, el SADE llega al óptimo mientras que el DE no se acerca en casi ninguna de ellas.

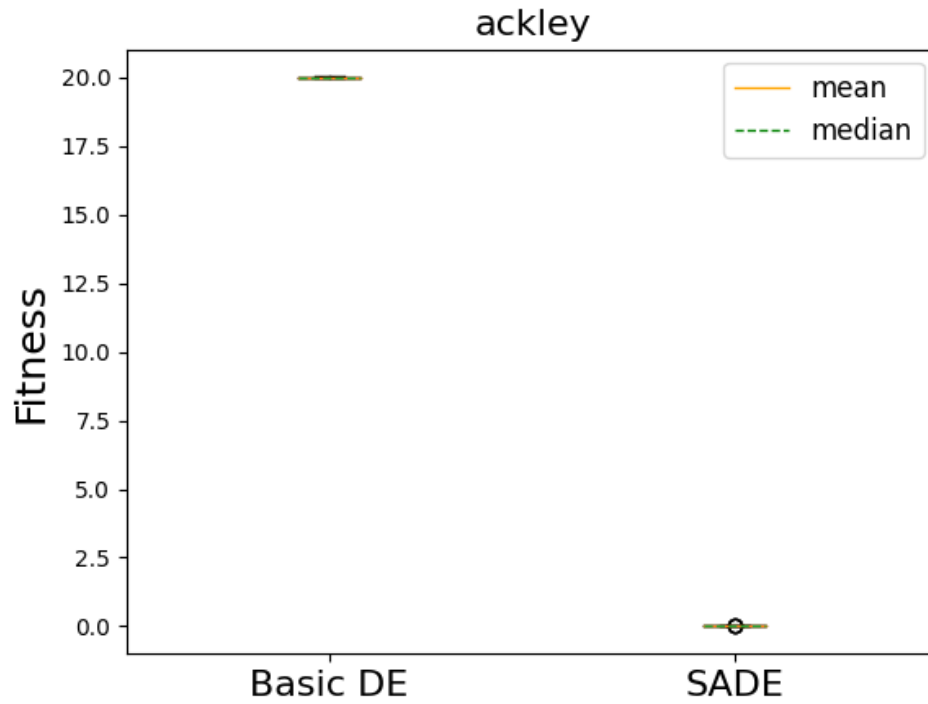


#### 4.3.1 Diagrama de cajas - Sphere



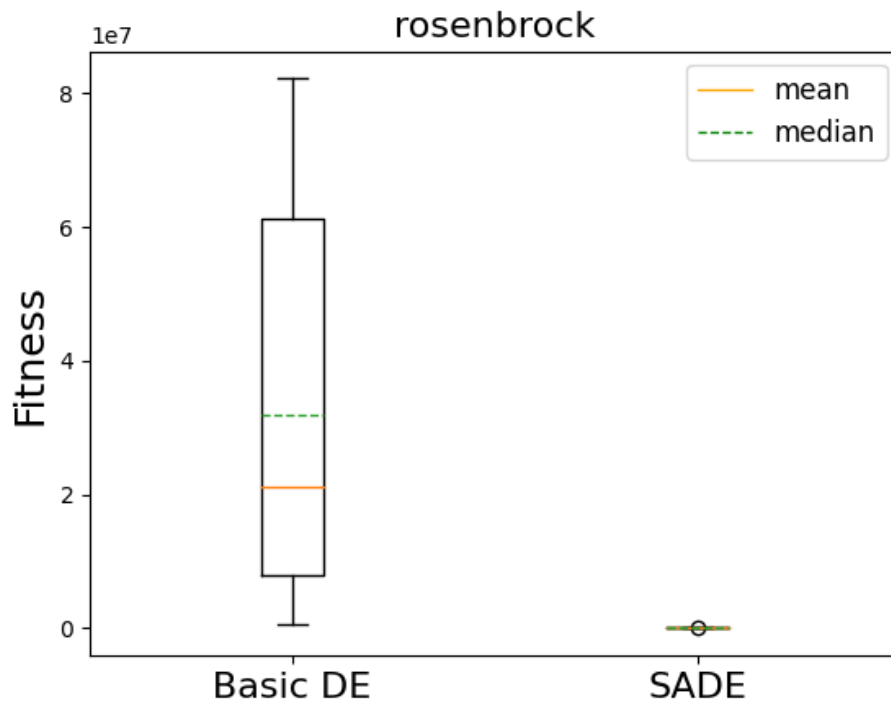
En la función *sphere*, se puede apreciar la diferencia de ambos algoritmos. Nuestro DE apenas se acerca al 0, con valores entre 3000 y 150, más o menos, mientras que el SADE llega perfectamente al óptimo con todos sus valores. De hecho, el diagrama de cajas del SADE no se aprecia porque todos sus valores son homogéneos como vimos en las tablas mostradas anteriormente en las que la *desv\_est* es casi 0.

#### 4.3.2 Diagrama de cajas - Ackley



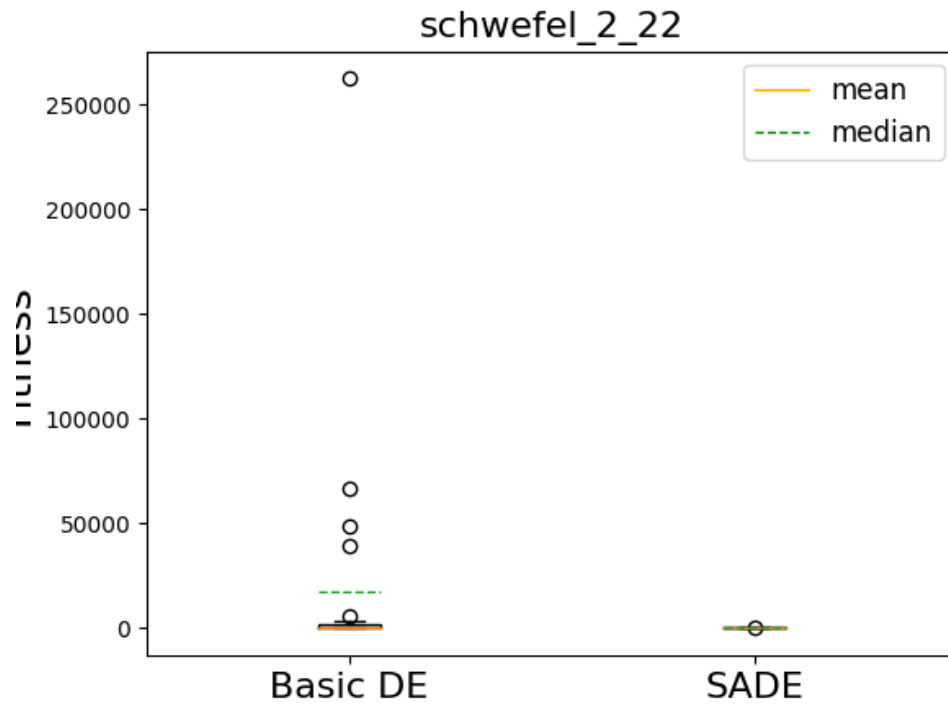
En esta imagen, que muestra los resultados de ambos algoritmos para la función *ackley*, podemos ver que se diferencian de 20. En este caso, el DE parece que alcanza un mínimo local ya que todos los valores están en 20, aunque se vea que, mediante el SADE, el mínimo global sea 0.

### 4.3.3 Diagrama de cajas - Rosenbrock



Esta figura muestra es muy representativa ya que el DE tiene valores muy superiores al SADE, del orden de millones (véase eje y). Este comportamiento es similar al encontrado anteriormente con la función *sphere* y a otras funciones de las que no vamos a poner diagrama porque todas se comportan prácticamente igual. Se puede seguir apreciando la diferencia de eficiencia de ambos algoritmos.

#### 4.3.4 Diagrama de cajas - Schwefel\_2\_22



Esta función es un caso aparte. Es la única que tiene valores más o menos parecidos tanto en DE como en SADE. Aunque el DE tiene gran parte de sus valores cercanos al óptimo (0), hay muchos valores atípicos muy alejados de este, lo que hace que la media se dispare y salga fuera del diagrama de cajas.

## 5 | Conclusión

De este trabajo hemos sacado varias conclusiones:

- Hay una gran diferencia de algoritmos de optimización que, por lo general, son mejores los ya implementados que los implementados por nosotros ya que son más complejos.
- Hemos descubierto formas de comparar algoritmos que nunca antes habíamos visto.
- Al usar diferentes librerías estadísticas, tanto para los test como para los diagramas de cajas, hemos podido adentrarnos un poco en este mundillo que es muy complejo y, si a eso se le suma  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , el aprendizaje es mayor.