Evolución Diferencial con perturbaciones Gaussianas

M. A. Sotelo-Figueroa[1], Arturo Hernández-Aguirre[2], Andrés Espinal[3] y J. A. Soria-Alcaraz[1]

[1] Universidad de Guanajuato, División de Ciencias Económico Administrativas, Departamento de Estudios Organizacionales, Guanajuato, Gto, México

[2] Centro de Investigación en Matemáticas, Departamento de Ciencias de la Computación, Guanajuato, Gto, México

[3] Tecnologico Nacional de México, Instituto Tecnológico de León, Departamento de Estudios de Posgrado e Investigación, León, Gto, México

masotelo@ugto.mx, artha@cimat.mx, andres.espinal@itleon.edu.mx, soajorgea@gmail.com

Resumen. La Evolución Diferencial es una metaheurística poblacional que ha sido ampliamente utilizada para la Optimización de problemas de Caja Negra. La muta es el operador principal de búsqueda en Evolución Diferencial, del cual hay diferentes esquemas reportados en el estado del arte; sin embargo dichos esquemas carecen de mecanismos para realizar una intensificación los cuales pueden permitir una mejor búsqueda y evitar óptimos locales. El presente artículo propone una perturbación Gaussiana con el objetivo de mejorar el desempeño de dos esquemas de muta bien conocidos y ampliamente utilizados en el estado del arte. Se realizaron pruebas en un conjunto de instancias de pruebas del CEC2013, los resultados obtenidos se compararon mediante la prueba no paramétrica de Friedman para determinar si el uso de la perturbación propuesta mejora el rendimientos de los esquemas originales.

Palabras clave: differential evolution, blackBox optimization, Gaussian perturbation.

1. Introducción

La Evolución Diferencial (ED) [21] es un algoritmo poblacional que ha sido ampliamente estudiado y básicamente se pueden dos líneas de investigación[5]: la búsqueda de nuevos esquemas de mutación con la finalidad de poder mejorar su rendimiento [2] [16] [1] [9] [12], y la otra es el estudio de los parámetros para obtener mejores resultados in [6] [19] [3] [14] [4].

Para hacer que la ED obtenga mejores resultados se ha hibridado con algunas otras técnicas como Búsqueda Local [18] u Optimización por Cumulo de Partículas [24]. También se han tratado de proponer nuevos esquemas de mutación como una Mutación Gaussiana [16]. Dichas propuestas tratan de mejorar el comportamiento de la ED sin embargo hacen que la ED se vuelva más compleja de lo que originalmente era.

Los problemas de caja negra [22] [8][11] [10] son problemas los cuales no se conoce a priori la función objetivo con la que se esta trabajando, motivo por el cual no se puede utilizar algoritmos basados en gradiente.

En el presente articulo se propone una perturbación Gaussiana la cual se puede utilizar con cualquier esquema de mutación para mejor el desempeño de la ED. Para probarla se utilizaron las 28 funciones propuestas en la sesión especial de Optimización Real de Parámetros del CEC2013[4]. Para discernir los resultados se aplico la prueba no paramétrica de Friedman.

El articulo se estructura como sigue: en la Sección 2 se presenta el algoritmo Evolución Diferencial y la perturbación Gaussiana que se propone, en la Sección 3 se explica el experimento que se realizo. Los resultados del experimento se muestran en la Sección 4. La Sección 5 muestra las conclusiones y el trabajo futuro.

2. Evolución diferencial

La Evolución Diferencial (ED) [21] fue desarrollada por R. Storm y K. Price en 1996. Es un algoritmo evolutivo el cual se basa en vectores y puede ser considerado como desarrollo adicional al Algoritmo Genético. La ED es un método de búsqueda estocástico con tendencias de auto organización sin necesidad de hacer uso la información de la derivada [23].

Como en los Algoritmos Genéticos [13], los parámetros a optimizar se representan como un vector d-dimensional y se aplican una serie de operadores sobre dichos vectores. Sin embargo, a diferencia del Algoritmo Genético, la ED aplica dicho operador sobre cada componente del vector.

Para un problema de optimización $d - dimensional$ con d parámetros, se genera una población inicial de n vectores donde se tienen x_i donde $i = 1, 2, \ldots, n$. Para cada solución x_i de cualquier generación t se puede usar la notación mostrada en la Ecuación 1 que consiste de $d - componentes$ en un espacio de búsqueda $d - dimensional$. Este vector puede ser considerado como el cromosoma o el genoma.

$$x_i^t = (x_1^t, x_2^t, \ldots, x_{d,i}^t) \tag{1}$$

La ED contiene tres pasos principales: mutación, cruza y selección. La mutación es llevada acabo por un Esquema de Mutación. Para cada vector x_i en cualquier tiempo o generación t, se generan tres números diferentes de manera aleatoria x_p, x_q y x_r en el tiempo t y se genera un vector de prueba mediante un Esquema de Mutación, como se ve en la Ecuación 2. El parámetro $F \in [0, 2]$ es denominado como peso diferencial.

$$v_i^{t+1} = x_p^t + F(x_p^t - x_r^t) \tag{2}$$

La cruza es controlada por la probabilidad $Cr \in [0, 1]$, y puede ser binomial o exponencial. El esquema de cruza binomial se aplica a cada uno de los componentes del vector mediante la generación de un número aleatorio uniformemente distribuido $r_i \in [0, 1]$ y se manipula como se muestra en la Ecuación 3 y se puede decidir en cada componente su se intercambia con el vector de prueba o no.

[4] http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/CEC2013.htm

$$u_{j,i}^{t+1} = \begin{cases} v_{j,i} & \text{si } r_i \leq C_r \\ x_{j,i} & \text{en otro caso} \end{cases} j = 1, 2, ..., d \tag{3}$$

En el esquema de cruza exponencial un segmento del vector de prueba es seleccionado empezando de la posición $k \in [0, d-1]$ y tamaño $L \in [1, d]$ de manera aleatoria. Este esquema se puede ver en la Ecuación 4.

$$u_{j,i}^{t+1} = \begin{cases} v_{j,i}^t \; v_{j,i}^t \text{ for } j = k, \ldots, k - L \in [1, d] \\ x_{j,i}^t \qquad \text{en otro caso} \end{cases} \tag{4}$$

La selección utilizada por la ED es en esencia la misma que se usa en el Algoritmo Genético. Dependiendo de lo que se esta haciendo se selecciona el valor más grande o el más pequeño. Dicha selección se puede ver en la Ecuación 5.

$$x_i^{t+1} = \begin{cases} u_i^{t+1} & \text{if } f(u_i^{t+1}) \leq f(x_i^t) \\ x_i^t & \text{en otro caso} \end{cases} \tag{5}$$

Los tres pasos de la ED se muestran en el Algoritmo 1. Cabe señalar que el uso de J es para asegurar que $v_i^{t+1} \neq x_i^t$ lo cual incrementa la eficiencia de la exploración. La eficiencia de la ED es controlada por dos parámetros: el peso diferencial F y la probabilidad de cruza C_r.

Algorithm 1 Algoritmo de Evolución Diferencial

Require: F peso diferencial, C_r probabilidad de cruza, n tamaño de la población.
Iniciar la población Inicial.
while no se cumpla el criterio de paro **do**
  for $i = 1$ to n **do**
    Para cada x_i se seleccionan de manera aleatoria 3 diferentes vectores x_p, x_r y x_r.
    Generar un nuevo vector v mediante algún esquema de mutación 2.
    Generar aleatoriamente un Índice $J_r \in \{1, 2, \ldots, d\}$.
    Generar aleatoriamente un número $r_i \in [0, 1]$.
    for $j = 1$ a n **do**
      Para cada parámetro $v_{j,i}$ (j-esimo componente de v_i), actualizarlo como:
      $u_{j,i}^{t+1} = \begin{cases} v_{j,i}^{t+1} & \text{si } r_i \leq C_r \text{ o } j = J_r \\ x_{j,i}^t & \text{si } r_i > C_r \text{ o } j \neq J_r \end{cases}$
    end for
    Seleccionar y actualizar la solución usando 5.
  end for
  Actualizar el contador $t = t + 1$
end while

En [17] se pueden ver 5 estrategias para mutar el vector v:

- DE/rand/1: $V_i = X_{r1} + F(X_{r2} - X_{r3})$.
- DE/best/1: $V_i = X_{best} + F(X_{r1} - X_{r2})$.
- DE/curren to to best/1: $V_i = X_i + F(X_{best} - X_i) + F(X_{r1} - X_{r2})$.

- DE/best/2: $V_i = X_{best} + F(X_{r1} - X_{r2}) + F(X_{r3} - X_{r4})$.
- DE/rand/2: $V_i = X_{r1} + F(X_{r2} - X_{r3}) + F(X_{r4} - X_{r5})$.

donde: r_1, r_2, r_3, r_4 son números enteros aleatoriamente generados y mutuamente diferentes, también deben de ser diferentes del vector de prueba i y del índice del mejor vector X_{best}.

2.1. Perturbación gaussiana

Con la propuesta de esta Perturbación Gaussiana se busca que la ED evolucione de manera normal bajo un determinado esquema de mutación y en caso de que el vector no se pueda mejorar mediante dichos pasos entonces se aplica una perturbación a cada dimensión tratando de que el vector actual pueda salir de un mínimo local. Esto se hace si la Ecuación 5 no se aplico y se quedo con el vector original. Lo que se hace es que para elemento del vector se le aplica una perturbación Gaussiana mediante la suma de un número generado aleatoriamente mediante una distribución Normal con media cero y una varianza de uno.

La Perturbación Gaussiana esta basada en un esquema de mutación Gaussiana[16] , la cual genera un vector de prueba usando una normal con media en un individuo, como se ve en la Ecuación 6, y con una varianza basada en la diferencia de dos elementos aleatorios. Este esquema de mutación esta utilizando el esquema Rand1 para hacer la generación del nuevo individuo.

$$u_i^{t+1} \sim N\left(\mu_i, \sigma_i^2\right) \tag{6}$$

$$\mu = x_{r1}^t \tag{7}$$

$$\sigma = \mid x_{r2} - x_{r3} \mid \tag{8}$$

Algorithm 2 Perturbación Gaussiana

```
1: if No se actualizo el Individuo usando la Ecuación 5 then
2:    for j = 1 to d do
3:       r ∈ N(0,1)
4:       u^{t+1}_{j,i} = x^t_{j,i} + r
5:    end for
6:    Seleccionar y actualizar la solución usando 5.
7: end if
```

3. Experimentos

Se utilizaron las 28 Funciones de la Tabla 1, las cuales fueron tomadas de la sesión especial de optimización de parámetros reales del CEC2013. Dichas funciones están para 2, 10, 30 y 50 dimensiones. Se ejecutaron 51 pruebas independientes para determinar la Mediana y la Desviación Estándar de cada una de las funciones. El máximo número de Llamadas a Función esta en función de la dimensión con la que se este trabajando y es de 10000*D; así pues para 2D=20000, 10D=100000, 30D=300000 y 50D=500000. El espacio de búsqueda para cada función es de $[-100, 100]^D$, la población se debe de inicializar de manera aleatoria uniformemente distribuida y se debe de utilizar el mismo método de optimización para todas las funciones.

Los parámetros utilizados por la ED fueron $F = 0.9$ y $Cr = 0.8$. Dichos parámetros se obtuvieron mediante un proceso de optimización de parámetros basado en Covering Arrays (CA) [20], los CAs se generaron mediante el Covering Array Library (CAS) [15] del National Institute of Standards and Technology (NIST) [5].

Para comparar los resultados obtenidos mediante los diferentes esquemas de mutación de ED se utilizo la prueba no paramétrica de Friedman [7] con un nivel de significancia del 99 % o un p-Valor menor que 0.1.

Tabla 1. Funciones del CEC2013 utilizadas como instancias de prueba

	No.	Función	$f_i^* = f_i(x^*)$
Funciones Unimodales	1	Sphere Function	-1400
	2	Rotated High Conditioned Elliptic Function	-1300
	3	Rotated Bent Cigar Function	-1200
	4	Rotated Discus Function	-1100
	5	Different Powers Function	-1000
Funciones Básicas Multimodales	6	Rotated Rosenbrock's Function	-900
	7	Rotated Schaffers F7 Function	-800
	8	Rotated Ackley's Function	-700
	9	Rotated Weierstrass Function	-600
	10	Rotated Griewank's Function	-500
	11	Rastrigin's Function	-400
	12	Rotated Rastrigin's Function	-300
	13	Non-Continuous Rotated Rastrigin's Function	-200
	14	Schwefel's Function	-100
	15	Rotated Schwefel's Function	100
	16	Rotated Katsuura Function	200
	17	Lunacek Bi_Rastrigin Function	300
	18	Rotated Lunacek Bi_Rastrigin Function	400
	19	Expanded Griewank's plus Rosenbrock's Function	500
	20	Expanded Scaffer's F6 Function	600
Funciones Compuestas	21	Composition Function 1	700
	22	Composition Function 2	800
	23	Composition Function 3	900
	24	Composition Function 4	1000
	25	Composition Function 5	1100
	26	Composition Function 6	1200
	27	Composition Function 7	1300
	28	Composition Function 8	1400

[5] http://csrc.nist.gov/groups/SNS/acts/index.html

4. Resultados

En las Tablas 2, 3, 4 y 5 podemos ver los resultados obtenidos para cada una de las diferentes dimensiones probadas. Los resultados que se muestran son la Mediana y la Desviación Estándar que se obtuvieron al aplicar cada uno de los esquemas de mutación a las diferentes función de prueba.

Los resultados para 2 dimensiones, Tabla 2, se puede observar que las Medias reportadas son prácticamente los óptimos de cada una de las funciones de prueba, y que la Desviación Estándar tiende a cero, lo que implica que en cada uno de las 51 pruebas independientes que se realizaron se llego casi al mismo resultado. Para el resto de las dimensiones, los resultados obtenidos con los diferentes esquemas de mutación para cada función objetivo varían.

Los valores de la prueba no paramétrica de Friedman aplicada a cada una de las dimensiones se puede ver en la Tabla 6, en el caso de la dimensión 2 es la única que el p-valor no es menor de 0.1 con lo cual no se puede determinar estadísticamente si hay algún esquema de mutación que tenga un desempeño diferente a los demás. Las otras dimensiones tienen p-Valores menores que 0.1 por lo que se realizo un procedimiento post-hoc [7] para determinar cual de los esquemas de mutación dieron mejores resultados.

La Tabla 7 contiene los resultados del procedimiento post-hoc de las dimensiones 20, 30 y 50. Dicha tabla contiene la posición en cada dimensión que ocupa cada uno de los esquemas usados, se muestran los resultados de manera ascendente mostrando con el número más pequeño aquel esquema que obtuvo el mejor desempeño.

5. Conclusiones y trabajos futuros

En base a los resultaos obtenidos en la Sección 4 podemos concluir lo siguiente:

- La Evolución Diferencial permite realizar la optimización de problemas de caja negra usando diferentes esquemas de mutación.
- Entre los diferentes esquemas de mutación encontrados en el estado del arte y probados en el presente articulo, el esquema Best1 es el que permite obtener mejores resultados.
- La perturbación Gaussiana propuesta permite ser incorporada a los diferentes esquemas de mutación, permitiendo en un espacio de búsqueda mayor mejorar los resultados que los esquemas por si solos obtienen.

Dentro de los trabajos futuros podemos considerar los siguientes:

- Usar números aleatorios con otro tipo de distribución que nos permitan intensificar y diversificar.
- Aplicar la Evolución Diferencial con la perturbación Gaussiana a otro tipo de problemas de optimización.

Agradecimientos. Los autores quieren agradecer a la Universidad de Guanajuato (UG) y al Instituto Tecnologico de León (ITL) por el apoyo brindado para poder realizar la presente Investigación.

Tabla 2. Resultados obtenidos en 2 Dimensiones

Función	Rand1		Rand1 Perturbación		Best1		Best1 Perturbación	
	Mediana	Desviación Estándar	Mediana	Desviación Estándar	Mediana	Desviación Estándar	Mediana	Desviación Estándar
1	-1400	0	-1400	0	-1400	0	-1400	0
2	-1300	0	-1300	0	-1300	0	-1300	0
3	-1200	0	-1200	0	-1200	0	-1200	0
4	-1100	0	-1100	0	-1100	0	-1100	0
5	-1000	0	-1000	0	-1000	0	-1000	0
6	-900	0	-900	0	-900	0	-900	0
7	-800	0	-800	14.500E-8	-800	0	-800	0
8	-689.3377	7.7096	-700	6.0700E-8	-700	2.6699	-700	0
9	-599.9790	10.162E-2	-600	1.2300E-4	-600	0	-600	0
10	-500	10.2545E-4	-500	16.9942E-4	-499.9926	94.6512E-4	-499.9926	80.4591E-4
11	-400	0	-400	0	-400	19.3129E-2	-400	0
12	-300	0	-300	0	-300	13.7949E-2	-300	0
13	-200	27.5899E-2	-200	38.6259E-2	-200	59.1737E-2	-200	96.2092E-2
14	-100	7.34525E-2	-100	6.05952E-2	-99.6878	5.3373	-99.6878	17.6587E-2
15	100	9.28302E-2	100	11.9007E-2	100.3122	6.5817	100.3122	5.6430
16	200.2055	10.580E-2	200.2837	11.9015E-2	200.4341	27.6091E-2	200.1446	19.4136E-2
17	300.0100	64.8657E-2	300.0264	48.4489E-2	300.0900	1.0324	302.0225	96.2327E-2
18	400	74.5786E-2	400	65.6815E-2	400	1.0307	402.0386	1.0155
19	500	0	500	0	500	53.0422E-4	500	30.3457E-4
20	600	82.4342E-4	600.0031	82.0231E-4	600.0194	84.6878E-4	600.0031	96.6389E-4
21	700	13.8648	700	0	700	27.7297	700	0
22	800	0	800	13.200E-14	800	6.1883	800	5.8100
23	900	0	900	4.3371	900	34.3032	900	20
24	1000	0	1000	74.0571E-2	1000	14.0500	1000	3.7075
25	1100	29.7368	1100	6.3700E-14	1100	48.1511	1100	0
26	1200	1.1224E-2	1200	2.78572E-2	1200.0810	20.1335E-2	1200.0810	1.0026
27	1400	50.0206	1300.4080	54.7587E-2	1400	40.9605	1300.8487	2.8579
28	1400	0	1400	0	1400	32.2190	1400	0

Tabla 3. Resultados obtenidos en 10 Dimensiones

Función	Rand1		Rand1 Perturbación		Best1		Best1 Perturbación	
	Mediana	Desviación Estándar	Mediana	Desviación Estándar	Mediana	Desviación Estándar	Mediana	Desviación Estándar
1	-1400	0	-1400	6.7800E-8	-1400	0	-1400	0
2	416306.31	214802.26	341737.85	135754.22	-1299.2456	24.5095	-923.3450	5172.0868
3	343002.02	636163.88	2541979.10	113E5	-1199.9424	2.2167	-1193.1031	72.7863
4	984.7043	1371.7921	-555.0403	211.2600	-1099.9988	44.3797E-4	-1098.9986	1.5002
5	-1000	1.3100E-10	-999.9999	45.500E-6	-1000	0	-1000	8.5700E-14
6	-899.9925	33.0102E-4	-899.5852	22.9179E-2	-900	93.8019E-2	-899.9999	55.2727E-2
7	-778.6989	11.0512	-787.3262	4.1405	-799.9973	2.2449	-799.8426	38.8166E-2
8	-679.6476	6.26167E-2	-679.6215	7.52653E-2	-679.5992	7.76678E-2	-679.6267	7.92253E-2
9	-590.9883	62.9856E-2	-592.2454	1.8715	-590.9685	87.1897E-2	-598.9312	1.8933
10	-499.3358	8.52432E-2	-499.1029	11.3437E-2	-499.8499	11.910E-2	-499.7758	20.1074E-2
11	-388.7206	2.7602	-380.2701	3.7180	-396.0202	2.3142	-397.0151	2.0714
12	-263.5439	7.3158	-257.2486	5.1130	-287.0655	7.7023	-285.0756	8.4215
13	-160.4450	5.1433	-158.1615	5.6212	-180.1075	9.9533	-169.6112	11.3929
14	1350.3154	267.3899	1030.1151	178.3200	168.5423	166.5590	87.0873	141.5081
15	2090.5090	156.9597	1112.2961	125.7929	1983.9856	244.4828	479.3273	200.3958
16	201.1530	18.5114E-2	201.1135	21.4722E-2	201.2751	24.7795E-2	201.1070	22.5713E-2
17	333.2465	5.1286	338.0571	4.2009	316.9115	3.2064	316.2581	3.1796
18	424.5464	3.0416	431.4449	4.6895	414.5922	2.7948	414.2326	2.3850
19	502.8092	46.5163E-2	503.1865	51.4913E-2	500.8897	42.7528E-2	501.0482	87.3303E-2
20	604.0924	16.3105E-2	602.4839	18.7561E-2	603.8289	32.8581E-2	602.3309	35.6801E-2
21	1100.1939	27.7566	1100.1939	34.500E-10	1100.1939	27.7566	1100.1939	68.200E-14
22	2546.6971	209.5217	2241.9218	185.8858	1233.4577	168.8049	1079.3355	279.1669
23	3053.8101	195.3891	2544.0314	158.6696	2912.9330	242.4062	2032.3326	490.0057
24	1224.6044	1.8565	1223.6982	1.4481	1223.2766	3.9215	1222.6930	15.3119
25	1323.9070	1.5264	1322.5013	1.4241	1323.0140	5.8763	1321.3607	3.1662
26	1427.0967	35.1082	1400.2457	9.10547E-2	1400.0157	44.4678	1400.0159	29.6483
27	1847.9867	17.0230	1839.8457	16.2134	1832.5907	24.1561	1836.7544	52.7770
28	1700	27.7297	1700.0121	48.6305	1700	47.0588	1700	93.5813

Tabla 4. Resultados obtenidos en 30 Dimensiones

Función	Rand1		Rand1 Perturbación		Best1		Best1 Perturbación	
	Mediana	Desviación Estándar	Mediana	Desviación Estándar	Mediana	Desviación Estándar	Mediana	Desviación Estándar
1	-1346.4059	38.0542	-1118.0943	141.0085	-1400	0	-1400	41.000E-8
2	112E7	277E6	979E5	167E5	659E5	703E5	332E5	148E5
3	2070E9	2000E9	1970E7	393E7	155E7	2520E7	716E6	139E7
4	342908.65	246484.92	29231.02	3264.3626	183170.40	105788.71	20698.94	3932.5735
5	-670.2021	138.4160	-653.5791	98.0488	-1000	18.300E-10	-999.9996	6.6800E-4
6	-852.3908	26.7010	-809.5061	36.7907	-882.7099	1.3664	-877.1137	1.6291
7	1042.5698	652.6231	-666.9936	12.1286	-450.9967	226.2352	-712.5903	30.1951
8	-679.0687	5.3800E-2	-679.0628	5.2371E-2	-679.0441	4.62247E-2	-679.0565	4.50616E-2
9	-560.2177	1.0239	-560.3796	1.0978	-560.1388	1.0592	-560.1474	1.0517
10	689.1837	741.5460	108.5105	192.1903	-499.9822	1.22565E-2	-498.7924	67.4287E-2
11	-188.9647	27.2281	-165.6650	19.8788	-363.1566	10.6429	-363.7414	11.2915
12	-12.5828	18.9520	-13.3075	17.5848	-71.7929	15.5653	-49.5101	22.9677
13	89.1128	18.3198	91.0993	15.6000	31.5563	18.9024	51.7046	16.4528
14	6747.4326	714.8719	6736.8623	494.1199	2251.4205	487.9502	1645.5606	522.4326
15	8999.9815	343	7167.6465	334.7258	8727.1068	347.4994	6979.3171	1200.1692
16	202.5787	22.6687E-2	202.4857	26.0314E-2	202.5192	23.6755E-2	202.4469	29.5087E-2
17	578.3695	24.3320	579.8762	21.2909	378.2476	12.8320	387.3654	11.1742
18	665.9283	25.7968	674.5000	23.2997	468.5144	8.9153	469.2101	10.4613
19	539.5813	25.4176	544.5058	21.1697	503.2208	2.5988	507.9325	4.7273
20	613.9282	11.1009E-2	611.6082	20.923E-2	613.7171	14.437E-2	611.4694	27.4063E-2
21	1237.5016	75.9275	1139.7762	87.4000	1000	57.6963	1000.0135	52.6740
22	8725.8409	772.3042	7951.9067	390.4807	3109.2959	504.2832	2531.5257	471.7557
23	10153.5262	395.2123	8545.5701	301.9254	9700.0509	331.1385	8517.9607	335.5761
24	1308.5288	2.5034	1302.5127	3.5268	1304.2682	3.2614	1301.5633	3.6273
25	1405.1394	3.1100	1399.0560	2.7703	1400.3797	3.1719	1399.8855	3.0722
26	1606.5142	17.9361	1530.9653	63.2531	1604.2630	9.8524	1405.7701	93.8747
27	2680.5745	34.1000	2616.3925	28.5592	2642.5637	30.4504	2603.9495	24.8560
28	2478.2177	185.2082	2491.3137	123.3221	1700	203.7669	1700.0346	64.4479

Tabla 5. Resultados obtenidos en 50 Dimensiones

Función	Rand1		Rand1 Perturbación		Best1		Best1 Perturbación	
	Mediana	Desviación Estándar	Mediana	Desviación Estándar	Mediana	Desviación Estándar	Mediana	Desviación Estándar
1	-1040.8523	235	-335.3356	585.4279	-1400	1.2100E-12	-1400	38.200E-6
2	321E7	876E6	305E6	483E5	20E7	185E6	854E5	539E5
3	20900E9	13800E9	4610E7	595E7	311E9	478E9	991E7	870E7
4	609558	295433.54	49059.94	4829.3853	404703.52	148341.93	41253.35	4874.5962
5	114.6304	474	-549.7202	111.7359	-1000	61.400E-8	-999.9907	90.9358E-4
6	-774.3136	41.1148	-684.7121	64.4253	-855.3971	8.0002	-853.5313	15.1197
7	2242.5014	1088.6112	-659.7467	9.2511	-52.6781	337.6897	-693.0075	19.0840
8	-678.8529	3.96501E-2	-678.8578	2.9287E-2	-678.8664	2.86874E-2	-678.8707	3.49506E-2
9	-526.6454	1.2908	-536.6595	6.3188	-526.7943	1.3887	-561.7574	13.9882
10	2234.6962	1097.0866	702.5670	283.5973	-499.1533	43.5963E-2	-496.3909	2.3646
11	12.2708	32.9327	36.3994	34.0209	-301.3726	24.2598	-286.8243	31.0356
12	233.1595	35.5541	243.6663	21.3201	154.9661	28.4237	187.8832	38.8346
13	329.0811	32.4769	353.1823	27.9270	257.3082	32.3157	315.7222	39.7084
14	11686.1406	1050.4285	12317.1506	784.0558	4961.4007	808.5805	3952.7790	735.6795
15	16504.6334	385.4880	14364.0033	378.4235	16120.3115	437.0822	14190.8173	365.0883
16	203.4003	24.1954E-2	203.4057	27.1238E-2	203.2891	30.5194E-2	203.3514	27.1652E-2
17	827.5058	38.5225	829.2380	38.4022	474.9279	24.1299	491.3891	26.3886
18	902.9463	48.8572	904.9828	45.0742	542.3676	25.1382	556.5838	27.9382
19	1340.5712	4400.3963	745.9946	402.1813	507.6820	2.4078	518.5668	9.2166
20	623.7504	16.6811E-2	620.9047	26.7632E-2	623.5780	21.0339E-2	620.8574	32.5134E-2
21	1569.8515	159.7738	1424.6311	219.5310	1000	53.3391	1100.0115	49.1729
22	14161.6903	1319.6516	14501.5546	865.3156	6366.1895	876.7257	5077.7514	784.6218
23	17421.5151	377.4880	15651.9869	323.7425	17214.1185	376.9616	15535.7750	457.3987
24	1397.6106	3.2987	1386.4351	3.9145	1392.4623	4.1267	1388.2413	3.5376
25	1490.2184	4.6573	1484.0797	2.8575	1486.8253	4.1214	1482.6281	3.3060
26	1698.1012	3.8452	1683.9892	22.5093	1693.3772	4.0488	1686.4911	42.1806
27	3560.2109	41.1272	3470.2224	36.1716	3503.1063	41.4514	3447.8082	41.3196
28	2377.4740	1672.4561	2243.1992	117.8650	1800	1277.5309	1800.0504	745.3048

Tabla 6. Valores y p-Valores obtenidos por la prueba no paramétrica de Friedman

	2D	10D	30D	50D
Valor	2.9678	35.7535	35.9464	45.6428
p-Valor	0.3966	8.4452E-8	7.6891E-8	7.2075E-10

Tabla 7. Clasificación de algoritmos mediante el procedimiento Post-hoc descrito en [7]

Algoritmo	10D	30D	50D
Best1 Perturbación	1.5893	1.5536	1.5357
Best1	1.9821	2.1250	1.9286
Rand1 Perturbación	3.1429	2.8214	2.8929
Rand1	3.2857	3.5000	3.6429

Referencias

1. Al-dabbagh, R., Botzheim, J., Al-dabbagh, M.: Comparative analysis of a modified differential evolution algorithm based on bacterial mutation scheme. In: Differential Evolution (SDE), 2014 IEEE Symposium on. pp. 1–8 (Dec 2014)
2. Bhowmik, P., Das, S., Konar, A., Das, S., Nagar, A.: A new differential evolution with improved mutation strategy. In: Evolutionary Computation (CEC), 2010 IEEE Congress on. pp. 1–8 (July 2010)
3. Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V.: Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. Evolutionary Computation, IEEE Transactions on 10(6), 646–657 (Dec 2006)
4. Brest, J., Maučec, M.: Self-adaptive differential evolution algorithm using population size reduction and three strategies. Soft Computing 15(11), 2157–2174 (2011)
5. Das, S., Suganthan, P.: Differential evolution: A survey of the state-of-the-art. Evolutionary Computation, IEEE Transactions on 15(1), 4–31 (Feb 2011)
6. Das, S., Konar, A., Chakraborty, U.K.: Two improved differential evolution schemes for faster global search. In: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation. pp. 991–998. GECCO '05, ACM, New York, NY, USA (2005)
7. Derrac, J., García, S., Molina, S., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation pp. 3–18 (2011)
8. Droste, S., Jansen, T., Wegener, I.: Upper and lower bounds for randomized search heuristics in black-box optimization. Theory of Computing Systems 39(4), 525–544 (2006)
9. Einarsson, G., Runarsson, T., Stefansson, G.: A competitive coevolution scheme inspired by de. In: Differential Evolution (SDE), 2014 IEEE Symposium on. pp. 1–8 (Dec 2014)
10. El-Abd, M.: Black-box optimization benchmarking for noiseless function testbed using artificial bee colony algorithm. In: Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation. pp. 1719–1724. GECCO '10, ACM, New York, NY, USA (2010)
11. El-Abd, M., Kamel, M.S.: Black-box optimization benchmarking for noiseless function testbed using particle swarm optimization. In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers. pp. 2269–2274. GECCO '09, ACM, New York, NY, USA (2009)
12. Fan, Q., Yan, X.: Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies. Cybernetics, IEEE Transactions on PP(99), 1–1 (2015)
13. Holland, J.: Adaptation in natural and artificial systems. University of Michigan Press (1975)
14. Jin, W., Gao, L., Ge, Y., Zhang, Y.: An improved self-adapting differential evolution algorithm. In: Computer Design and Applications (ICCDA), 2010 International Conference on. vol. 3, pp. V3–341–V3–344 (June 2010)
15. Kacker, R.N., Kuhn, D.R., Lei, Y., Lawrence, J.F.: Combinatorial testing for software: An adaptation of design of experiments. Measurement 46(9), 3745 – 3752 (2013)
16. Li, D., Chen, J., Xin, B.: A novel differential evolution algorithm with gaussian mutation that balances exploration and exploitation. In: Differential Evolution (SDE), 2013 IEEE Symposium on. pp. 18–24 (April 2013)
17. Luke, S.: Essentials of Metaheuristics. Lulu (2009)
18. Noman, N., Iba, H.: Accelerating differential evolution using an adaptive local search. Evolutionary Computation, IEEE Transactions on 12(1), 107–125 (Feb 2008)
19. Omran, M., Salman, A., Engelbrecht, A.: Self-adaptive differential evolution. In: Hao, Y., Liu, J., Wang, Y., Cheung, Y.m., Yin, H., Jiao, L., Ma, J., Jiao, Y.C. (eds.) Computational

Intelligence and Security, Lecture Notes in Computer Science, vol. 3801, pp. 192–199. Springer Berlin Heidelberg (2005)
20. Rodriguez-Cristerna, A., Torres-Jiménez, J., Rivera-Islas, I., Hernandez-Morales, C., Romero-Monsivais, H., Jose-Garcia, A.: A mutation-selection algorithm for the problem of minimum brauer chains. In: Batyrshin, I., Sidorov, G. (eds.) Advances in Soft Computing, Lecture Notes in Computer Science, vol. 7095, pp. 107–118. Springer Berlin Heidelberg (2011)
21. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. of Global Optimization 11, 341–359 (December 1997)
22. Wang, G., Goodman, E., Punch, W.: Toward the optimization of a class of black box optimization algorithms. In: Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on. pp. 348–356 (Nov 1997)
23. Yang, X.S.: Nature Inspired Metaheuristic Algorithms. Luniver Press, 2da edn. (2008)
24. Zavala, A., Aguirre, A., Diharce, E.: Particle evolutionary swarm optimization algorithm (peso). In: Computer Science, 2005. ENC 2005. Sixth Mexican International Conference on. pp. 282–289 (2005)