

Estudio comparativo de algoritmos metaheurísticos basados en evolución diferencial y estimación de distribuciones probabilísticas para la optimización de funciones en el espacio continuo.

Marlene Lopez, Gustavo López

Abstract—Los algoritmos metaheurísticos son útiles para optimizar funciones, i.e. encontrar máximos o mínimos. Mientras que la Evolución Diferencial es un algoritmo metaheurístico basado en poblaciones que evolucionan al aplicarles ciertos operadores específicos, los algoritmos de estimación de probabilidad (como UMDA y BUMDA) se basan en una distribución para guiar tendencias de mejora de actitud en la función objetivo. Muy diversas funciones bien conocidas sirven como referentes indicativos de la efectividad de algún algoritmo que intenta optimizarles; en este trabajo se utilizan 5 de tales funciones para evaluar la capacidad de optimización de los algoritmos metaheurísticos Evolución Diferencial, UMDA y BUMDA a través de una comparativa en sus implementaciones y un análisis estadístico significativo.



INTRODUCCIÓN

En todas áreas en donde se busque solucionar algún problema en específico es factible la búsqueda de la mejor solución posible / deseable. Esta búsqueda ha originado el desarrollo de algoritmos de optimización, que pueden ser de naturaleza determinista o estocástica. Los primeros requieren enormes esfuerzos computacionales para resolver problemas de optimización, que tienden a fallar a medida que aumenta el tamaño del problema. Esta es la motivación para emplear algoritmos de optimización estocástica bio-inspirados como alternativas computacionalmente eficientes al enfoque determinista. Las metaheurísticas como la Evolución Diferencial se basan en la mejora de una población de soluciones y emplean mayormente operadores como cruce, mutación y selección para resolver el problema dado. Por otro lado, los algoritmos de estimación de probabilidad (EDAs, por sus siglas en inglés) guían

la búsqueda de la solución óptima construyendo y muestreando modelos probabilísticos de soluciones candidatas prometedoras. En general, los algoritmos metaheurísticos gozan de una buena reputación para resolver problemas de optimización para cualquier función del espacio continuo.

En el presente trabajo se muestra la implementación del algoritmo metaheurístico Evolución Diferencial, y los EDAs UMDA y BUMDA para la búsqueda de los óptimos globales de las funciones Ackley, Griewank, Rosenbrock, Schwefel y Zakharov.

ANTECEDENTES

Los Algoritmos Evolutivos son sistemas computacionales diseñados para solucionar problemas altamente no-lineales, tomando ideas del proceso evolutivo y de adaptación de la naturaleza, las cuales se agrupan bajo la denominada teoría Neo-Darwiniana de la evolución [1]

Gran parte de la terminología utilizada en los algoritmos evolutivos se ha tomado prestada de los conceptos biológicos en los que se inspira (por ejemplo, cromosoma, alelos, etc.).

- Marlene Lopez is with Alumno de la maestría en Ciencias de la Computación, Instituto Tecnológico de León
- Gustavo López is with Alumno de la maestría en Ciencias de la Computación, Instituto Tecnológico de León.

Una característica importante de los algoritmos evolutivos es que son poblacionales, lo que significa que manipulan simultáneamente un conjunto de soluciones potenciales al problema y no una solución única, como suelen hacer las técnicas clásicas de optimización. El uso de una población de soluciones potenciales a un problema hace a los algoritmos evolutivos menos susceptibles de quedar atrapados en óptimos locales [2].

Los operadores genéticos básicos son los siguientes:

- * Selección: Consiste de un mecanismo (probabilístico o determinista) que permite elegir a los individuos que fungirán como padres de la siguiente generación.
- * Cruza o Recombinación: Se refiere al intercambio de información (material genético) entre dos padres que han sido seleccionados con base en su aptitud.
- * Mutación: Consiste en hacer pequeñas perturbaciones a los individuos recién creados para la nueva población con la finalidad de explorar zonas del espacio de búsqueda que la cruce no pueda alcanzar.

MÉTODOS

Evolución Diferencial

La Evolución Diferencial es un método de optimización que funciona a través de la mejora iterativa de una solución candidata con respecto a una medida de calidad. Mantiene una población de soluciones candidatas y crea nuevas combinando las existentes con reglas simples y manteniendo aquellas que resultan ser las mejores [3].

Algoritmos de Estimación de Distribución (EDAs)

UMDA

El UMDA (Univariate Marginal Distribution Algorithm, Algoritmo de distribución marginal univariado) consiste en la probabilidad de selección 'S', obtenida después de que los puntos en el espacio de muestra completo hayan sido ordenados según los requisitos. Luego se generará una población de soluciones y se combinará con la probabilidad de selección [ühlenbein2001?hnenbein2001?] Estos pasos de

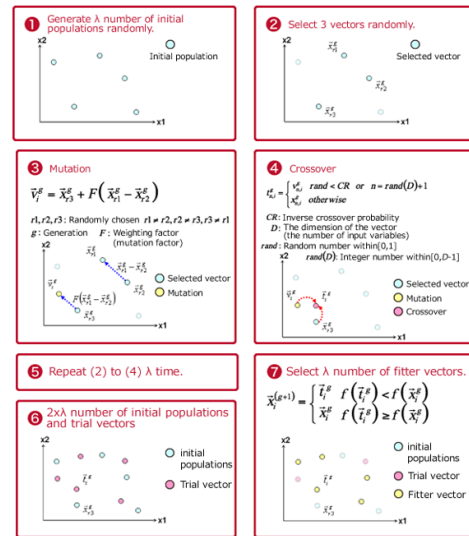


Figure 1. Algoritmo de la Evolución Diferencial

UMDA se realizan iterativamente hasta que se satisfaga la función objetivo, y la solución se elige para ser el individuo con la mejor condición física en toda la población.

UMDA

STEP 0: Set $t \leftarrow 1$. Generate $N \gg 0$ points randomly.

STEP 1: Select $M \leq N$ points according to a selection method. Compute the marginal frequencies $p^i(x_i, t)$ of the selected set.

STEP 2: Generate N new points according to the distribution $p(x, t+1) = \prod_{i=1}^n p^i(x_i, t)$. Set $t \leftarrow t+1$.

STEP 3: If termination criteria are not met, go to STEP 1.

Figure 2. Pseudocódigo del UMDA

BUMDA

Con el objetivo de guiar efectivamente la convergencia hacia el valor óptimo, el algoritmo BUMDA es respaldado por dos elementos clave: (1) la distribución de Boltzmann y (2) la aplicación de un método de truncamiento de la población en los mejores individuos.

RESULTADOS

Para la experimentación se utilizó una población de 100 vectores de 10 dimensiones inicializados con una distribución uniforme en el dominio de cada función. Para la Evolución Diferencial se realizaron dos variaciones del

BUMDA

1. Give the parameter and stopping criterion:
nsample ← Number of individuals to be sample.
minvar ← minimum variance allowed.
2. Uniformly generate the initial population P_0 , set $t = 0$.
3. While $v > \text{minvar}$ for all dimensions
 - (a) $t \leftarrow t + 1$
 - (b) Evaluate and truncate the population according algorithm in Figure 1.
 - (c) Compute the approximation to μ and v (for all dimensions) by using the selected set (of size n_{selec}), and Equations (6) and (9), as follows:

$$\mu \approx \frac{\sum_{i=1}^{n_{selec}} x_i g(x_i)}{\sum_{i=1}^{n_{selec}} g(x_i)}, \quad v \approx \frac{\sum_{i=1}^{n_{selec}} g(x_i)(x_i - \mu)^2}{1 + \sum_{i=1}^{n_{selec}} g(x_i)},$$
 where $g(x_i) = g(x) - g(x_{n_{selec}}) + 1$.
 Note: the individuals can be sorted to simplify the computation, and $g(x_{n_{selec}})$ is the minimum (for maximization case) objective value of the selected individuals.
 - (d) Generate $n_{sample} - 1$ individuals from the new model $Q(x, t)$, and insert the elite individual.
4. Return the elite individual as the best approximation to the optimum.

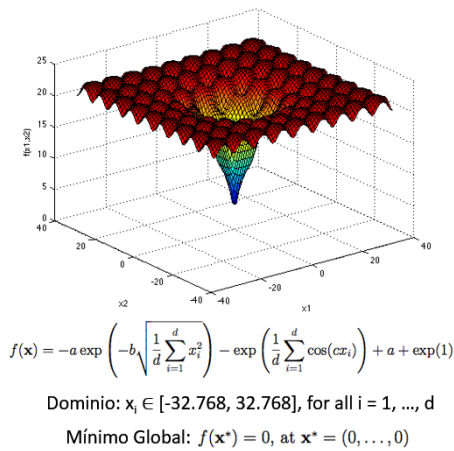


Figure 3. Seudocódigo del BUMDA

Figure 4. Función Ackley

Table 1
Configuraciones de los algoritmos utilizados

	Máximo Generaciones	Máximo de llamadas a función	Probabilidad de cruza	Valor F	Selección	Media	Mediana	Desviación Estándar	Mínimo	Máximo
UMDA	1,000	500,000	-	-	BUMDA	-2.34E-12	-4.44E-15	1.32E-11	-7.58E-11	-8.88E-11
BUMDA	1,000	500,000	-	-	DE1	1.29E-02	9.93E-03	7.90E-03	3.43E-03	4.11E-02
DE1	1,000	500,000	0.8	0.9	DE2	8.52E-10	8.05E-10	3.13E-10	3.74E-10	1.60E-09
DE2	1,000	500,000	0.3	0.9	UMDA	5.22E-05	4.44E-15	3.00E-04	8.88E-16	1.72E-03

Table 2
Resultados Ackley

algoritmo con parámetros diferentes. Dada la naturaleza del BUMDA caracterizado para la maximización de funciones y el enfoque de minimización de funciones de este trabajo, se ajustó el algoritmo para cumplir con este cometido. En la tabla 1 se denotan las configuraciones de los algoritmos utilizados.

A continuación se denota por cada función utilizada los resultados obtenidos, así como algunas observaciones del comportamiento de los experimentos.

Función Ackley

La función Ackley, su dominio, su óptimo global y una representación en dos dimensiones están denotados en la siguiente figura. [4]

La tabla 2 muestra los resultados obtenidos en las ejecución de 33 experimentos de cada uno de los algoritmos previamente descritos para la función Ackley en diez dimensiones.

Se puede notar que el UMDA dio los mejores resultados para esta función, seguido del BUMDA.

Función Griewank

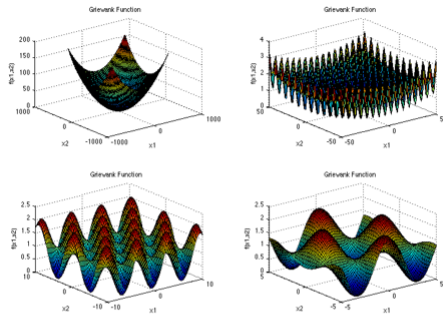
La función Griewank, su dominio, su óptimo global y una representación en dos dimensiones están denotados en la siguiente figura.

La tabla 3 muestra los resultados obtenidos en las ejecución de 33 experimentos de cada uno de los algoritmos previamente descritos para la función Griewank en diez dimensiones.

En este caso, tanto el UMDA como el BUMDA llegaron a la solución óptima, mien-

Table 3
Resultados Griewank

	Media	Mediana	Desviación Estándar	Mínimo	Máximo
BUMDA	-1.95E-05	0	7.44E-05	-3.81E-04	0
DE1	4.16E-01	4.12E-01	5.91E-02	2.63E-01	5.16E-01
DE2	1.47E-04	8.10E-05	1.94E-04	8.08E-07	9.99E-04
UMDA	2.47E-05	0	1.42E-04	0	8.14E-04



$$f(\mathbf{x}) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

Dominio: $x_i \in [-600, 600]$, for all $i = 1, \dots, d$

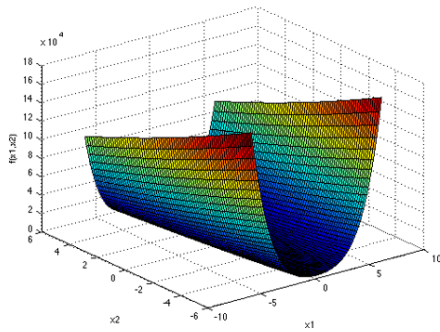
Mínimo Global: $f(\mathbf{x}^*) = 0$, at $\mathbf{x}^* = (0, \dots, 0)$

Figure 5. Función Griewank

tras que las variaciones del algoritmo de Evolución Diferencial también llegaron a aproximaciones aceptables.

Función Rosenbrock

La función Rosenbrock, su dominio, su óptimo global y una representación en dos dimensiones están denotados en la siguiente figura.



$$f(\mathbf{x}) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$$

Dominio: $x_i \in [-5, 10]$, for all $i = 1, \dots, d$

Mínimo Global: $f(\mathbf{x}^*) = 0$, at $\mathbf{x}^* = (1, \dots, 1)$

Figure 6. Función Rosenbrock

La tabla 4 muestra los resultados obtenidos en las ejecución de 33 experimentos de cada uno de los algoritmos previamente descritos para la función Rosenbrock en diez dimensiones.

Table 4
Resultados Rosenbrock

	Media	Mediana	Desviación Estándar	Mínimo	Máximo
BUMDA	-8.13	-8	3.90E-01	-9.42	-7
DE1	1.24	1.13	3.70E-01	4.71E-01	2.09
DE2	4.33	4.40	8.89E-01	2.16	6.26
UMDA	8.23	8	5.62E-01	7	1.10E+01

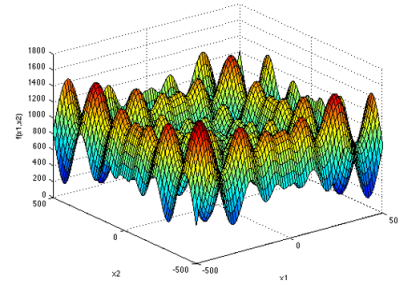
Table 5
Resultados Schwefel

	Media	Mediana	Desviación Estándar	Mínimo	Máximo
BUMDA	9.14E+03	7478	5.41E+03	2.55E+03	24750
DE1	1.33E+03	1.29E+03	1.56E+02	1.00E+03	1.62E+03
DE2	7.71E+02	7.90E+02	1.52E+02	4.50E+02	9.97E+02
UMDA	-1.65E+04	-10501	1.47E+04	-65494	-2.74E+04

Para esta función, los EDAs entregaron resultados más pobres en la búsqueda del mínimo, mientras que la variación DE1 del algoritmo de Evolución Diferencial proveyó la aproximación más adecuada.

Función Schwefel

La función Schwefel, su dominio, su óptimo global y una representación en dos dimensiones están denotados en la siguiente figura.



$$f(\mathbf{x}) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|})$$

Dominio: $x_i \in [-500, 500]$, for all $i = 1, \dots, d$

Mínimo Global: $f(\mathbf{x}^*) = 0$, at $\mathbf{x}^* = (420.9687, \dots, 420.9687)$

Figure 7. Función Schwefel

La tabla 5 muestra los resultados obtenidos en las ejecución de 33 experimentos de cada uno de los algoritmos previamente descritos para la función Schwefel en diez dimensiones.

Table 6
Resultados Zakharov

	Media	Mediana	Desviación Estándar	Mínimo	Máximo
BUMDA	-1.09E+01	-10	8.96	-4.57E+01	-1
DE1	1.04	9.61E-01	5.92E-01	1.77E-01	2.77
DE2	3.70E-01	3.36E-01	1.78E-01	8.69E-02	8.69
UMDA	1.97E+01	17	1.37E+01	2	5.25E+01

Table 7
Resultados de las pruebas no paramétricas

Prueba	P-Valor	Nivel Significancia	H ₀
Friedman	0.037403	0.05	Se rechaza
Friedman Alineado	0.294867	0.05	No se rechaza
Quade Test	0.001308	0.05	Se rechaza

Debido a la complejidad de esta fórmula, i.e. la gran cantidad de mínimos locales que contiene, el desempeño de esta experimentación resultó ser la menos plausible de este trabajo, siendo los EDAs los que entregaron una mejor aproximación (que sigue estando lejos del óptimo global).

Función Zakharov

La función Zakharov, su dominio, su óptimo global y una representación en dos dimensiones están denotados en la siguiente figura.

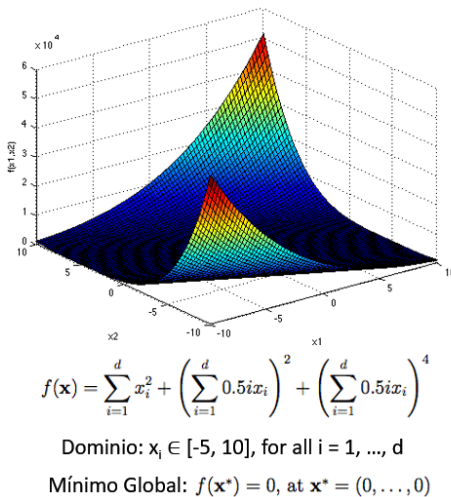


Figure 8. Función Zakharov

La tabla 6 muestra los resultados obtenidos en las ejecución de 33 experimentos de cada uno de los algoritmos previamente descritos para la función Zakharov en diez dimensiones.

El algoritmo de Evolución Diferencial en su variación DE2 dio los mejores resultados en este caso, superando a los resultados entregados por los EDAs.

Dada que no fue notorio la prevalencia de algún algoritmo de dominar en la optimización

de las funciones utilizadas, se procedió a realizar un estudio estadístico no paramétrico con la finalidad de esclarecer de manera significativa el comportamiento de los algoritmos.

Pruebas Estadísticas no paramétricas

Las pruebas estadísticas no paramétricas son recomendables cuando no se puede asumir que los datos experimentales se ajustan a alguna distribución conocida. Con el dato estadístico de la mediana, se aplicaron sobre las experimentaciones las pruebas Friedman, Friedman alineado y Quade con ayuda de la herramienta KEEL [5], bajo las siguientes hipótesis:

- H_0 - Los algoritmos propuestos tienen comportamientos idénticos.
- H_1 - Al menos un par de algoritmos tienen comportamientos diferentes entre sí.

Se obtuvieron los resultados denotados en la tabla 7.

La siguiente figura muestra los *rankings* obtenidos por las pruebas no paramétricas realizadas, y que denotan con el menor valor aquel algoritmo que presenta mayor diferencia en su comportamiento, i.e., que tiene un mejor rendimiento.

CONCLUSIONES

- La optimización de cualquier función dada es un ejercicio específico y particular de cada problema, por lo que es muy difícil encontrar un algoritmo (y sus configuraciones) que resulte útil para optimizar todas las funciones.
- Con el desarrollo de pruebas estadísticas con un nivel de significancia del 0.05, el

Algorithm	Ranking	Algorithm	Ranking
BUMDA	3.9	BUMDA	16.3
DE1	1.8	DE1	9.6
DE2	2.4	DE2	9.4
UMDA	1.9	UMDA	6.7

Ranking Friedman Ranking Friedman
Alineado

Algorithm	Ranking
BUMDA	3.9333
DE1	2
DE2	2.6
UMDA	1.4667

Ranking Quade

Figure 9. Rankings de las pruebas no paramétricas

algoritmo UMDA (con las configuraciones establecidas) obtuvo un mejor rendimiento que el algoritmo DE1, DE2, BUMDA para los problemas de aproximación de mínimos conocidos en las funciones denotadas en este trabajo.

REFERENCES

- [1] D. B. Fogel, "Introduction to evolutionary computation," in *Evolutionary Computation* 1. IOP Publishing Ltd. [Online]. Available: <https://doi.org/10.1887/2F0750306645%2Fb877c1>
- [2] N. C. Cortes and C. A. C. Coello, "Sistema inmune artificial para solucionar problemas de optimización," Ph.D. dissertation, PhD thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2004.
- [3] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec 1997. [Online]. Available: <https://doi.org/10.1023/A:1008202821328>
- [4] "Optimization Test Functions and Datasets," <https://www.sfu.ca/~ssurjano/optimization.html>, accessed on Thu, May 24, 2018. [Online]. Available: <https://www.sfu.ca/~ssurjano/optimization.html>
- [5] J. Alcala-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework," vol. 17, pp. 255–287, 01 2010.