

# **I Datos Generales**

Titulo del proyecto:  
Ackley function optimization .

Nombre del alumno:  
Ing. Iván Alejandro García Amaya.

Numero de expediente:  
290712.

Programa de Estudios:  
Maestría en Ciencias Inteligencia Artificial.

Nombre del Doctor:  
Dr. Marco Antonio Aceves Fernández.

Materia:  
Computo evolutivo.

## II Introducción

La función Ackley se usa ampliamente para probar algoritmos de optimización. En su forma bidimensional, se caracteriza por una región externa casi plana y un gran agujero en el centro. La función plantea el riesgo de que los algoritmos de optimización, particularmente los algoritmos de escalada, queden atrapados en uno de sus muchos mínimos locales. (Bäck & Back, 1996)

Valores recomendados para  $a = 20$ ,  $b = 0.2$ ,  $c = 2\pi$

$$f(x) = -a \exp\left(-b \sqrt{1/d \sum x_i^2}\right) - \exp(1/d \sum \cos(cx_i)) + a + \exp(1) \quad (1)$$

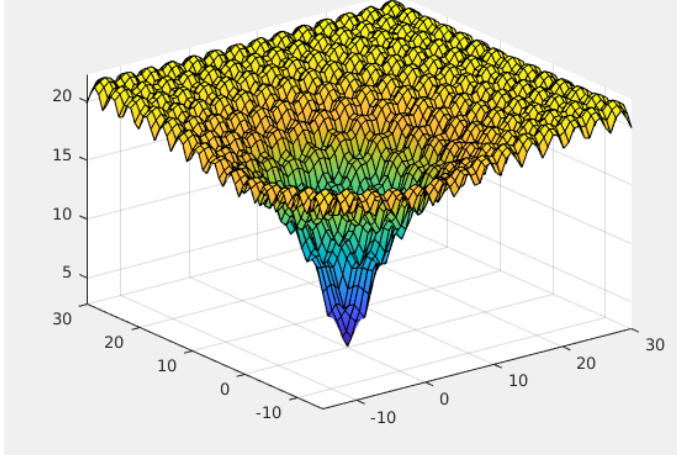


Figure 1: Ackley function plot 3d.

### III Marco Teórico.

#### *Populations.*

Una población es una colección de individuos. Una población consta de una cantidad de individuos que se están probando, los parámetros del fenotipo que definen a los individuos y alguna información sobre el espacio de búsqueda. Los dos aspectos importantes de la población utilizados en los algoritmos genéticos son: (Sivanandam & Deepa, 2007).

- La generación de población inicial.
- El tamaño de la población.

#### *Objective Function.*

Una función objetivo constituye el objetivo de un problema de optimización. Este objetivo podría maximizarse o minimizarse eligiendo variables o variables de decisión que satisfagan todas las restricciones del problema. La conveniencia de un conjunto de variables como una posible solución a un problema de optimización se mide por el valor de función objetivo correspondiente a un conjunto de variables. (Bozorg-Haddad, Solgi & Loáiciga, 2017)

#### *Fitness function.*

El valor de la función objetiva no siempre es la medida elegida de la conveniencia de una solución. Por ejemplo, el algoritmo puede emplear una forma transformada de la función objetivo mediante la adición de sanciones que evitan la violación de las restricciones, en cuyo caso la función transformada se llama *the fitness function*. *The fitness function* se emplea para evaluar la conveniencia de posibles soluciones (Bozorg-Haddad, Solgi & Loáiciga, 2017).

#### *Diversity.*

La diversidad se refiere a las diferencias entre los individuos, que pueden estar en los niveles de genotipo o fenotipo. Es ampliamente aceptado dentro de la comunidad de Evolutionary Algorithms (EA) que la alta diversidad de una población contribuye en gran medida al desempeño del EA. (Črepinšek, Liu & Mernik, 2013).

“El progreso en la evolución depende fundamentalmente de la existencia de variación de la población.” (McPhee & Hopper, 1999)

Desafortunadamente, un problema clave en muchos sistemas de Computación Evolutiva (EC) es la pérdida de diversidad a través de la convergencia prematura. Esta falta de diversidad a menudo conduce al estancamiento, ya que el sistema se encuentra atrapado en los óptimos locales, sin la diversidad genética necesaria para escapar ”. (Črepinšek, Liu & Mernik, 2013).

Existen muchas medidas diferentes para la diversidad, tanto genotípicas como fenotípicas, pero no existe una medida única que se adapte a todos los problemas y a los diferentes tipos de EA.

Una población diversa es un requisito previo para la exploración a fin de evitar la convergencia prematura a los óptimos locales. Por otro lado, promoviendo la diversidad en todas las etapas de un proceso evolutivo podría incluso ser contraproducente en una fase donde se necesita una alta explotación. (Črepinšek, Liu & Mernik, 2013).

La relación entre diversidad y exploración y explotación aún no está clara, y se necesita más investigación, especialmente al identificar los tipos (fenotípicos / genotípicos) y cantidades de diversidad en diferentes etapas evolutivas (Burke, 2004).

Como ya se mencionó, la diversidad se puede medir en tres niveles.

- Nivel de genotipo (estructural / sintáctico / genotípico): diferencias entre genomas dentro de una población.
- Nivel de fenotipo (conductual / semántico / fenotípico): diferencias entre los valores de aptitud física dentro de una población.
- Una medida compleja o compuesta: una combinación de los dos casos anteriores.

#### *Selection.*

La selección es el proceso de elegir dos padres de la población para cruzar. Después de decidir sobre una codificación, el siguiente paso es decidir cómo

realizar la selección, es decir, cómo elegir individuos en la población que crearán descendencia para la próxima generación y cuántos descendientes creará cada uno. El propósito de la selección es enfatizar a los individuos más en forma en la población con la esperanza de que sus hijos tengan mayor aptitud física, los cromosomas se seleccionan de la población inicial para ser padres para la reproducción. El problema es cómo seleccionar estos cromosomas. Según la teoría de la evolución de Darwin, los mejores sobreviven para crear una nueva descendencia. (Sivanandam & Deepa, 2007).

- Roulette Wheel Selection.
- Random Selection.
- Rank Selection.
- Tournament Selection.
- Boltzmann Selection.

#### *Crossover.*

Este ocurre entre dos parejas de soluciones. *The crossover* genera dos nuevos hijos que es el producto de los gen de los padres, en otras palabras, una nueva solución recibe algunas variables de decisión de una solución principal y el resto de la otra solución principal. Goldberg(1989) y muchalewicz(1996) han descrito varios métodos de cruce, incluyendo.

- *Crossover de punto.*
- *Crossover de dos punto.*
- *Crossover uniforme.*

#### *Single Point Crossover.*

El algoritmo genético tradicional utiliza el cruce de un solo punto, donde los dos se aparean. Los cromosomas se cortan una vez en los puntos correspondientes y se intercambian las secciones después de los cortes. Aquí, un punto cruzado o de cruce se selecciona aleatoriamente a lo largo de la longitud de las cadenas y bits acoplados al lado de los sitios cruzados se intercambian. Si se elige el sitio apropiado, se pueden obtener mejores hijos combinando

buenos padres, de lo contrario, obstaculiza gravemente la calidad de la generación. Los bits al lado del punto de cruce se intercambian para producir hijos. El punto de cruce se puede elegir al azar (Sivanandam & Deepa, 2007).

*Elitism.*

El primer mejor cromosoma o los pocos mejores cromosomas se copian a la nueva población. El resto se hace de manera clásica. Tales individuos pueden perderse si no son seleccionados para reproducirse o si el cruce o la mutación los destruyen. Esto significativamente mejora el rendimiento de la GA. (Sivanandam & Deepa, 2007).

*Termination criteria.*

Cada iteración de un algoritmo termina con una nueva solución. El algoritmo evalúa the fitness function de cada solución y se mueve a la siguiente iteración, o termina si *the remination criteria* es satisfecho.

## IV Materiales y Métodos.

### IV.I Materiales.

El algoritmo se programa usando Matlab.

### IV.II Métodos.

*Roulette Wheel Selection (RWS).*

En este método de selección proporcional, a las soluciones se les asignan números proporcionales a sus valores de aptitud. Si la aptitud promedio de todos los miembros de la población es  $f_{avg}$  una solución con una aptitud  $f_i$  obtiene un número esperado de copias  $f_i/f_{avg}$ , la implementación de este método de selección como su nombre lo dice se considera como una ruleta mecanismo, donde la rueda se divide en N divisiones (tamaño de la población), donde el tamaño de cada uno está marcado en proporción a la aptitud de cada miembro de la población. El individuo con mayor valor de aptitud, se espera que sea seleccionado por la rueda de ruleta (RWS) y sera elegido con mayor frecuencia.

Entonces usando el valor de aptitud  $f_i$  de toda la población, se puede obtener la probabilidad de seleccionar la i-ésima cadena que es  $p_i = f_i / \sum_{j=1}^n f_{avg}$  entonces, el acumulado de la probabilidad es  $p_i = \sum_{j=1}^i P_j$ . (Bozorg-Haddad, Solgi & Loáiciga, 2017)

probabilidad = probabilidad previa + aptitud/ la suma de la aptitud general.

*Two point crossover*

Consiste en tomar una matriz identidad de manera aleatoria del cromosoma 1, A1, intercambiarlo con la matriz identidad de A2 que la igual es tomada de manera aleatoria., generando como resultado O1 y O2.

$$A1 = [1,0,—0,1,—0];$$

$$A2 = [1,1,—1,0,—1];$$

$$O1 = [1,0,—1,0,—0];$$

$$O2 = [1, 1, -0, 1, -1];$$

#### *Scramble Mutation.*

Para la esta mutación usualmente se selecciona el 10% de la población, este porcentaje puede ser contenido por los los individuos menos aptos, seleccionados de manera aleatoria.

$$A = [1, 2, 3, 4, 5, 6, 7, 8] .$$

Se seleccionan los cortes a mutar, generando la matriz identidad.

$$A = [1, 2|3, 4, 5, 6|7, 8] .$$

Se retira de la cadena principal, la subcadena.

$$A = [1, 2|x, x, x, x|7, 8] .$$

$$A_1 = [3, 4, 5, 6] .$$

Generando dos cromosomas resultantes.

$$A = [1, 2, 7, 8] .$$

$$A_1 = [3, 4, 5, 6] .$$

Se realiza la acción de *Scramble* a la cadena cortada con cuidando para evitar que los *Gens* no queden en la posición original, después se reinserta la cadena en una posición diferente de donde se tomo de manera aleatoria.

$$A_1 = [1, 4, 6, 3, 5, 7, 8] .$$

#### Mutación Heurística.

Para este método, se genera de manera aleatoria un corte, en el cromosoma, generando una matriz llamada matriz identidad.

$$A = [1, 2, 3, |4, 5, 6|, 7, 8]$$



Por medio de permutación se comparan las posibles combinaciones, una vez obtenida la mejor, la matriz identidad se reintegra a su posición, dando como resultado u individuo más apto.

$$A = [1, 2, 3, 6, 5, 4, 7, 8]$$

## V Pseudocódigo y Diagrama de flujo

### V.I Pseudocódigo.

---

**Algorithm 1:** Travelling Salesman Problem

---

**input** : Random poblacion of the size  $l$

**output:** A approximate solution

1 *special treatment of the first line;*

2 **for**  $i \leftarrow 0$  to  $l$  **do**

3     **for**  $j \leftarrow 0$  to  $w$  **do**

4          $pred1, pred2 \leftarrow sel\_pad\_best\_float\_Ackley;$

5          $desc1, desc2 \leftarrow two\_point\_crossover\_float\_Ackley;$

6     **end**

7      $sort\ and\ evaluate\ generation \leftarrow objective\_function\_float\_Ackley;$

8      $best\ generation \leftarrow biology\_competition\_float\_Ackley;$

9     **if**  $a$  is equal to  $b$  **then**

10          $mutated\ generation \leftarrow Heuristic\_Mutation\_float\_Ackley;$

11     **end**

12 **end**

---

## V.II Diagrama de flujo.

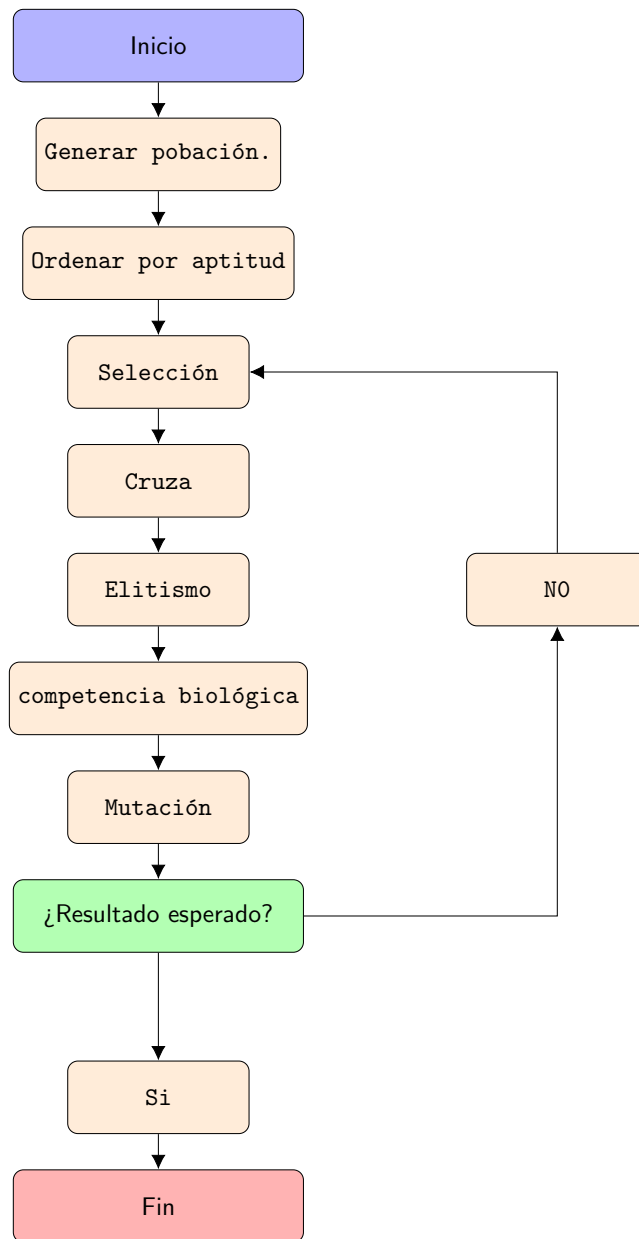


Figure 2: Diagrama de flujo.

## VI Desarrollo.

Población.

La población se generó compuesta por cromosomas de 82 genes, dentro de la función  $[cell\_pob] = poblation\_worm\_float\_Ackley(number\_of\_solution)$ , con números binarios 0 y 1. Donde de los primeros 41 números binarios representan a  $x1$ , siendo los 41 restantes la representación de  $x2$  ..

$$cromosoma = [a_1, a_2, a_3, \dots, a_n];$$

Donde  $a \in \{0, 1\}$ , y  $n = 82$ .

*Objective function.*

Dentro de la función  $[cell] = objective\_function\_float\_Ackley(cell)$  se evalúa la función  $f(x1, x2)$ , para obtener la aptitud de la solución que es la inversa de la función.

$$x1bin = [a_1, a_2, a_3, \dots, a_i]; \text{ donde } i = 41.$$

$$x2bin = [a_{42}, a_{43}, a_{44}, \dots, a_j]; \text{ donde } j = 82.$$

$$x1 = x1bin * pow2(n - 1 : -1 : -m).'; \text{ . } n = 16, m = 25.$$

$$x2 = x2bin * pow2(n - 1 : -1 : -m).';$$

Valores recomendados para  $a = 20$ ,  $b = 0.2$ ,  $c = 2\pi$

$$f(x) = a + \exp(1) - a * \exp(-b \sqrt{1/d(x_1^2 + x_2^2)}) - \exp(1/d(\cos(cx_1) + \cos(cx_2))) \quad (2)$$

$$x1, x2 \in \{-15, 30\}.$$

Generaciones.

Dentro de las generaciones a crear, nos encontramos con el llamado de la mayoría de las funciones implementadas, en cada ciclo creamos una nueva generación, haciendo uso de diferentes funciones, que se describen a continuación.

Comenzando con la función *sel\_ pad\_ best\_ float\_ Ackley*, dentro de esta función se hace uso de la función *my\_ own\_ RWS\_ best\_ float\_ Ackley*, como su nombre lo dice, utiliza la técnica Roulette Wheel Selection, basada en la probabilidad, selecciona a los nuevos padres, tomando en cuenta, que entre más aptos sean estos, mayor sera la probabilidad de que sean seleccionados, pero es necesario mencionar, que en problemas donde la población es grande, la probabilidad disminuye tanto entre todos los individuos, que nos encontramos con una técnica casi aleatoria de selección, siendo esta una desventaja de esta técnica.

Cruza.

Dentro de la función  $[desc1, desc2] = two\_point\_crossover\_float\_Ackley(pred1, pred2)$ , como su nombre lo menciona, se usa el método de cruza two point cross over, que consiste en tomar una matriz identidad de manera aleatoria del pred1 he intercambiarla con la matriz identidad de un pred2 que la igual es tomada de manera aleatoria.

$$pred1, pred2 \rightarrow cromosoma = [a_1, a_2, a_3, \dots, a_n]; .$$

$$x1a = [a_1, a_2, a_3, \dots, a_i]; .$$

$$x2a = [a_{42}, a_{43}, a_{44}, \dots, a_j]; .$$

$$x1b = [a_1, a_2, a_3, \dots, a_i]; .$$

$$x2b = [a_{42}, a_{43}, a_{44}, \dots, a_j]; .$$

Se toma de manera aleatoria la matriz identidad, de x1a y x2a, que es insertada de manera aleatoria en x1b, x2b y de sentido opuesto.

$$matrizidentidad = [a_k, a_2, a_3, \dots, a_l]; . \text{ donde } k = \text{random y } l = k+8.$$

Competencia Biológica.

La función *biology\_ competition\_ float\_ Ackley* acepta en total una población del doble de la generación actual, comprendida, por esta misma y la anterior, es necesario tomar en cuenta varios conceptos, como lo es la diversidad, para mantener la diversidad en la generación, se manda llamar la función *delete\_repeated\_float\_Ackley* que a su vez manda llamar a la función *scramble\_met\_*

*per\_one\_float - Ackley*, que como su nombre lo menciona, tiene el propósito de eliminar soluciones repetidas por medio de la técnica de mutación Scramble, evitando así una pérdida de diversidad, lo cual podría derivar en una convergencia prematura, dando como resultado una estagnación local.

“El progreso en la evolución depende fundamentalmente de la existencia de variación de la población.” (McPhee & Hopper, 1999).

Las posibles soluciones mutadas (soluciones repetidas) y las demás soluciones, son ordenadas por aptitud con ayuda de la función *objective\_function\_worm*, dando como resultado de la función *biology\_competition - Ackley* a los individuos con mayor aptitud.

#### Mutación Heurística.

Después de cada 10 generaciones se manda llamar la función *heuristic\_mutation\_float - Ackley*, tomando en cuenta el concepto de explotación, se muta el 5 por ciento de la población, se toma esta acción con la finalidad de encontrar mejores soluciones dentro de las primeras 100 generaciones, pero se aplica cada 10 generaciones ya que es un método un tanto agresivo, pero se hace pretendiendo tener un buen balance entre explotación y exploración.

Dentro de esta función se manda llamar a la función *permu\_loc\_float - Ackley*, es encargada de seleccionar de manera aleatoria una matriz identidad del individuo, después de una combinatoria de 720 posibles soluciones, se selecciona la de mayor aptitud.

## VII Resultados.

En la figura 3, se muestra la tabla de resultados, por generaciones, es necesario mencionar que en algunos casos se llega a una mejor respuesta, en menos generaciones.

Generación	f(x)	x1x2
100	3.37175166897907e-07	-1.19209289550781e-07 , 0
15	4.21469076528069e-07	1.49011611938477e-07 , 0
250	8.42937186718018e-08	2.98023223876953e-08 , 0
350	1.07086675633639e-05	8.94069671630859e-08 , - 3.7848949432373e-06
450	8.42937186718018e-08	2.98023223876953e-08 , 0
550	8.42937186718018e-08	-2.98023223876953e-08 , 0

Figure 3: Tabla resultados por generación .

En la figura 4, se muestra el aumento de la aptitud por generación, podemos ver un aumento bastante agresivo en un determinado momento esto por la naturaleza del problema.

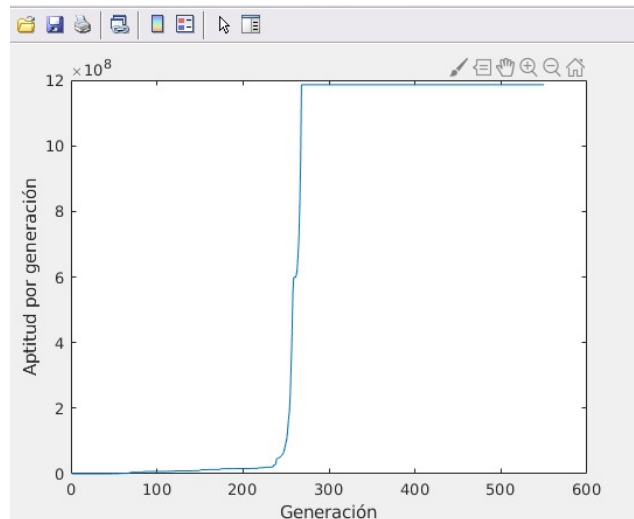


Figure 4: Aptitud por generación.

En la figura 5 se muestra  $f(x)$  por generación, en un momento se vuelve imperceptible el cambio de la pendiente.

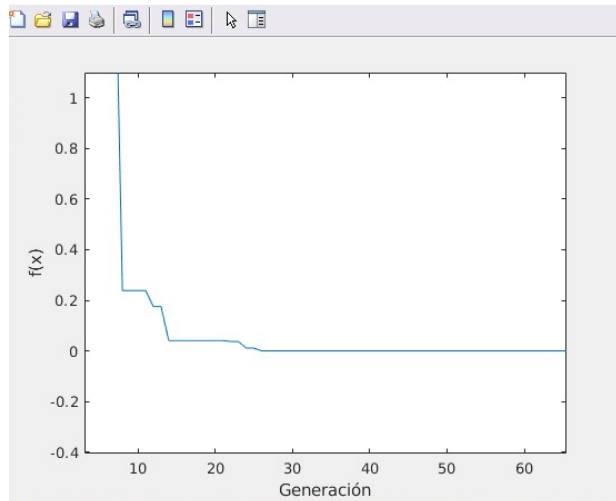


Figure 5:  $f(x)$  por generación.

En la figura 6 se muestra el plot en 3d, además de las posibles soluciones después de las 11 generaciones, pero son solo con números enteros, en la figura 7, se muestra la solución después de 550 generaciones con 20 decimales como resultado de  $f(x) = 8.42937186718018e-08$  y  $x_1x_2 = -2.98023223876953e-08$ , 0



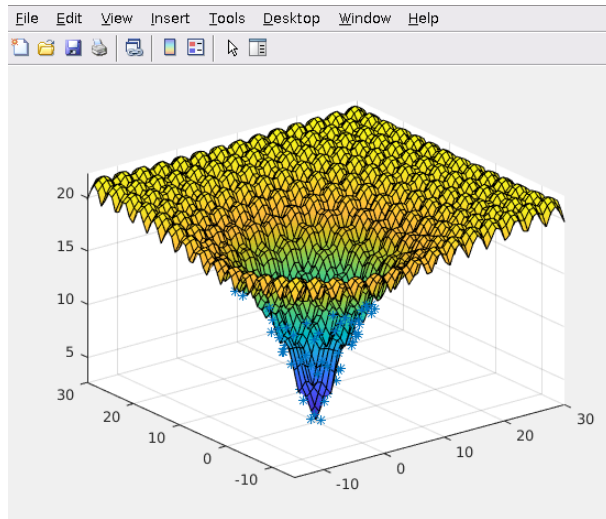


Figure 6: Resultados de números reales.

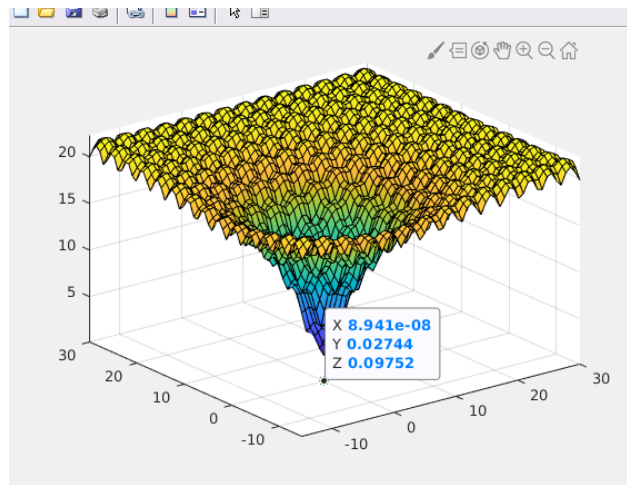


Figure 7: Resultados de numeros con 20 decimales.

## VIII Discusión de los Resultados.

La naturaleza del problema muestra uno de los problemas importantes en optimización, que es caer en un mínimo local, la función de Ackley, cuenta con una gran cantidad de mínimos locales, además de demostrar interés en la importancia de la exactitud de los números decimales, el primer acercamiento al problema se hizo de manera más superficial, haciéndolo con números enteros, el segundo y enfoque final se hizo con números con 20 decimales, buscando una aproximación mas completa.

En las figuras 4 y 5, se observa, que tenemos una estagnación, como se menciono, el mínimo global se encuentra rodeado por una gran cantidad de mínimos locales, cuando logramos entrar en la circunferencia del mínimo global es cuando se observa el aumento abrupto en la eficiencia, después se observa un aparente estagnación, pero es necesario mencionar, que en este punto, el aumento se enfoca principalmente en los dígitos e-07, por ende el aumento puede no ser observado en estos puntos.

## IX Conclusión.

La correcta aproximación de la solución es en gran medida, dependiente, de una correcta y distribuida exploración y explotación, tomando en consideración que para una buena exploración es necesaria una buena diversidad en nuestras generaciones, evitando de esta manera una convergencia prematura y por ende una estagnación en un optimo local.

Se menciona que el problema de optimización Ackley, es una prueba para la optimización ya que es complicado abordarlo con los métodos clásicos de optimización, abordándolo con Computo Evolutivo, en este caso con Algoritmos Genéticos, es en gran medida una técnica fuerte, ya que en este específico caso, el aplicar dos métodos de mutación, nos ayuda a enfrentar el problema de estagnación, siendo el método Scramble de gran ayuda para la exploración y el método Heurístico para la explotación.

## X Bibliografía.

Bäck, T., & Back, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. New York, EUA: Oxford University Press.

Bonomi, E., & Lutton, J. (1984). The N-City Travelling Salesman Problem: Statistical Mechanics and the Metropolis Algorithm. *SIAM Review*, 26(4), 551–568. <https://doi.org/10.1137/1026105>

Bozorg-Haddad, O., Solgi, M., & Loáiciga, H. A. (2017). *Meta-heuristic and Evolutionary Algorithms for Engineering Optimization*. Pondicherry, India: Wiley.

Črepinšek, M., Liu, S., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms. *ACM Computing Surveys*, 45(3), 1–33. <https://doi.org/10.1145/2480741.2480752>

Deep, Kusum, & Mebrahtu, Hadush (2011). New Variations of Order Crossover for Travelling Salesman Problem. *International Journal of Combinatorial Optimization Problems and Informatics*, 2(1), 2-13. [fecha de Consulta 12 de Marzo de 2020]. ISSN: . Disponible en: <https://www.redalyc.org/articulo.oa?id=2652/265219618002>.

Fogel, D. B.. (1988). *An Evolutionary Approach to the Traveling Salesman Problem*. Springer-Verlag, 60(60), 139–144.

Sivanandam, S. N., & Deepa, S. N. (2007). *Introduction to Genetic Algorithms*. Pvt., India: Springer Berlin Heidelberg.