

Universidad San Francisco de Quito

Nombre: José Gabriel García

Código: 00211322

NRC: 3322

Tarea 2

1) Genetic Algorithm (GA)

- **Solución a la función de Beale:** El par ordenado que minimiza la función es $x = 3$, $y = 0.5$ con $f(x, y) = 0$. La solución encontrada por este algoritmo en 100 iteraciones fue la siguiente:

```
Epoca 99
Valor de f: 1.770982917195897e-12
Par ordenado: [3.00000095, 0.49999997]
```

- **Travelling Salesman Problem:** La solución a este problema se muestra en la pregunta 5.

2) Particle Swarm Optimization Algorithm (PSO)

- **Solución a la función de Beale:** La solución exacta es la presentada anteriormente, y la obtenida con este algoritmo con 100 iteraciones fue:

```
Valor de X: 3.012500984550519
Valor de Y: 0.5033219135977989
Valor de la funcion: 2.599953685049689e-05
```

- **Solución para la función de Rastrigin para $A = 10$, y $n = 10$:** Esta es una función vectorial, y recibe como argumento un vector de dimensión n . Su valor mínimo se da cuando el vector que recibe es igual al vector cero, $\vec{0}$. La función de Rastrigin es:

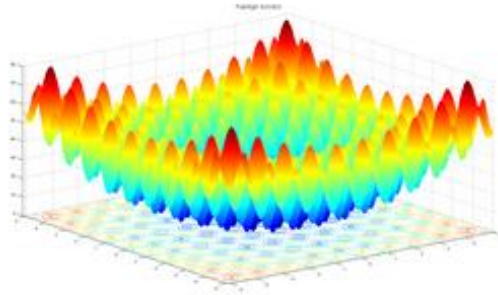
$$f(\vec{x}) = An + \sum_{i=1}^n [x_i^2 + A \cos(2\pi x_i)]$$

Con $x_i \in [-5.12, 5.12]$

Así, para minimizarla se utilizó el algoritmo PSO con 500 iteraciones y la mejor respuesta encontrada tras varios intentos fue:

```
Valores del vector X:
[ 0.9163201843280868
-0.039538108036015485
0.06246424724807409
0.1326113992786775
-0.9352074743920966
-1.0971607602595348
-0.10638806507458876
0.013426853859791033
0.8219287748335867
-1.185984180417347 ]
Valor de la funcion: 27.257269435834502
```

En este algoritmo cada partícula es un vector, y ya que se tiene un vector de partículas, entonces se está recorriendo una matriz, por lo que la complejidad del algoritmo es de $O(n^2)$, y realizar muchas iteraciones lo puede hacer lento, pero se puede optar por utilizarlo varias veces hasta encontrar una buena solución. En este caso, la mejor fue la indicada anteriormente, y como se puede observar el vector \vec{x} tienden al vector $\vec{0}$, el cual es la solución. Por otro lado, es complicado encontrar el mínimo de esta función ya que tiene varios mínimos por lo que es posible que las partículas tiendan a un mínimo local en vez del global. A continuación, se presenta una gráfica de esta función para visualizar su comportamiento y visualizar que contiene varios mínimos locales.



3) Ant System Algorithm (ACO-AS):

- **Travelling Salesman Problem:** La solución a este problema se muestra en la pregunta 5.

4) Artificial Bee Colony Optimization (ABC)

- **Solución a la función de Beale:** La solución exacta es la presentada anteriormente, y la obtenida con este algoritmo con 100 iteraciones fue:

```
Mejor solución para X: 3.0007470230319897
Mejor solución para Y: 0.5001834807298229
Valor de la función: 8.931360527303227e-08
```

Este algoritmo puede hacerse más eficiente cambiando el valor del límite al cual se reinicia una posible solución para converger de manera más rápida y no expandir soluciones que seguramente no conllevaran al valor buscado. Para este caso se utilizó un límite de 50 que puede ser un poco grande. Sin embargo, para un límite de 10, la solución con 100 iteraciones fue:

```
Mejor solución para X: 2.99999999967153
Mejor solución para Y: 0.4999999999568499
Valor de la función: 5.109924100570883e-20
```

La cual demuestra ser mucho mejor que la obtenida con el algoritmo genético y el algoritmo PSO. De esta manera, podría utilizarse este algoritmo para optimizar la función de Rastrigin de manera más rápida colocando un límite cercano al cero. Finalmente, la comparación de los resultados de optimizar la función de Beale se resumen en la siguiente tabla en la cual todos los algoritmos tuvieron 100 iteraciones.

Algoritmo usado	Valor de x	Valor de Y	Valor de la función
AG	3.00000095	0.49999997	1.771e-12
PSO	3.01250098	0.50332191	2.600e-5
ABC	2.99999999	0.49999999	5.101e-20

5) Comparar las soluciones de Travelling Salesman Problem GAs vs. ACO-AS

Todos los algoritmos utilizan 500 iteraciones y se mide el tiempo de ejecución.

50 Nodos			
Algoritmo Usado	Camino Encontrado	Tiempo requerido [s]	Distancia
ACO	[1, 25, 10, 14, 35, 4, 36, 9, 5, 32, 45, 46, 49, 48, 16, 42, 8, 2, 12, 44, 19, 39, 24, 17, 30, 47, 18, 15, 37, 29, 22, 43, 27, 13, 21, 28, 38, 6, 34, 33, 40, 26, 7, 41, 20, 31, 3, 50, 23, 11, 1]	30.78	248.00
GA	[32, 7, 39, 25, 16, 48, 20, 36, 42, 29, 49, 17, 45, 38, 35, 15, 1, 26, 41, 9, 5, 27, 2, 40, 44, 0, 19, 8, 47, 4, 3, 28, 30, 24, 13, 18, 10, 43, 46, 12, 11, 34, 6, 33, 21, 22, 31, 14, 23, 37, 32]	1.12	205.49
ABC	[47, 42, 22, 17, 25, 9, 20, 29, 19, 21, 39, 33, 48, 26, 8, 36, 40, 2, 37, 28, 45, 14, 0, 12, 27, 30, 32, 11, 24, 35, 4, 18, 44, 49, 6, 10, 34, 7, 5, 43, 41, 13, 38, 23, 15, 3, 16, 31, 46, 1, 47]	3.01	322.00

100 Nodos			
Algoritmo Usado	Camino Encontrado	Tiempo requerido [s]	Distancia
ACO	[1, 63, 73, 61, 58, 74, 99, 47, 6, 30, 39, 11, 84, 95, 81, 71, 68, 38, 19, 97, 72, 50, 78, 48, 42, 31, 79, 92, 96, 80, 57, 36, 21, 23, 25, 34, 82, 41, 2, 85, 98, 77, 76, 26, 3, 59, 60, 12, 54, 65, 55, 83, 35, 4, 62, 53, 44, 27, 5, 100, 37, 75, 45, 40, 18, 49, 89, 87, 67, 28, 52, 32, 70, 17, 33, 20, 22, 93, 86, 13, 56, 14, 15, 88, 43, 9, 24, 69, 8, 66, 94, 91, 16, 51, 29, 7, 46, 64, 10, 90, 1]	151.61	465.00
GA	[0, 58, 40, 71, 5, 13, 30, 28, 92, 90, 42, 79, 18, 52, 43, 60, 82, 88, 33, 22, 91, 32, 75, 14, 83, 31, 23, 16, 39, 61, 51, 59, 66, 20, 96, 17, 38, 80, 77, 7, 21, 85, 37, 56, 50, 9, 2, 76, 1, 62, 81, 74, 35, 15, 45, 73, 41, 27, 49, 48, 67, 84, 25, 6, 70, 64, 46, 11, 68, 97, 65, 53, 34, 55, 12, 93, 24, 63, 10, 57, 87, 95, 78, 94, 89, 4, 47, 72, 3, 19, 69, 86, 8, 54, 99, 36, 26, 98, 29, 44, 0]	2.31	462.59
ABC	[86, 82, 74, 32, 84, 56, 1, 79, 3, 34, 95, 38, 91, 98, 58, 66, 90, 39, 40, 64, 49, 96, 73, 75, 21, 0, 85, 44, 10, 83, 19, 99, 97, 80, 52, 67, 28, 89, 87, 69, 24, 63, 50, 4, 51, 59, 16, 76, 42, 46, 13, 23, 62, 93, 14, 60, 30, 36, 81, 17, 27, 43, 22, 9, 53, 45, 29, 71, 33, 57, 26, 8, 15, 20, 61, 18, 35, 78, 54, 2, 88, 6, 77, 11, 68, 70, 47, 37, 72, 31, 65, 48, 5, 25, 92, 12, 55, 41, 94, 7, 86]	11.97	648.00

150 Nodos			
Algoritmo Usado	Camino Encontrado	Tiempo requerido [s]	Distancia
ACO	[1, 140, 66, 28, 68, 14, 29, 81, 63, 17, 148, 22, 3, 143, 25, 13, 114, 91, 16, 130, 128, 121, 51, 20, 58, 119, 85, 21, 67, 78, 118, 46, 146, 69, 132, 93, 83, 99, 7, 123, 34, 90, 52, 111, 98, 102, 44, 150, 75, 36, 126, 8, 31, 43, 103, 87, 101, 56, 97, 142, 57, 104, 147, 127, 40, 136, 2, 35, 124, 30, 60, 125, 108, 45, 11, 88, 100, 55, 109, 33, 70, 89, 64, 129, 59, 112, 144, 4, 50, 41, 82, 145, 23, 139, 53, 37, 6, 32, 113, 65, 73, 120, 86, 77, 84, 149, 5, 92, 9, 131, 138, 110, 38, 96, 133, 135, 47, 71, 107, 10, 48, 116, 95, 27, 24, 137, 72, 19, 18, 26, 54, 79, 94, 117, 39, 49, 15, 74, 134, 62, 42, 61, 80, 76, 122, 105, 141, 106, 12, 115, 1]	351.57	723.00
GA	[47, 3, 72, 8, 86, 18, 138, 117, 93, 144, 12, 46, 87, 60, 39, 127, 64, 65, 68, 6, 70, 53, 104, 34, 55, 101, 11, 94, 24, 30, 129, 123, 110, 130, 78, 145, 98, 132, 73, 36, 20, 27, 5, 113, 51, 41, 67, 61, 105, 91, 134, 66, 112, 76, 80, 81, 1, 133, 114, 0, 10, 95, 119, 126, 148, 82, 88, 97, 9, 26, 2, 116, 139, 146, 149, 111, 142, 33, 56, 35, 141, 17, 15, 106, 42, 79, 13, 92, 140, 118, 69, 124, 115, 19, 58, 109, 29, 49, 107, 59, 45, 108, 50, 120, 37, 7, 77, 85, 16, 84, 21, 128, 121, 136, 143, 131, 23, 22, 14, 137, 25, 135, 83, 147, 32, 96, 38, 74, 62, 75, 31, 48, 71, 63, 28, 100, 90, 122, 40, 54, 99, 44, 125, 57, 43, 52, 89, 103, 4, 102, 47]	4.17	795.74
ABC	[122, 138, 81, 140, 111, 13, 67, 133, 57, 7, 109, 90, 2, 124, 141, 9, 48, 68, 19, 75, 123, 55, 33, 73, 97, 89, 70, 4, 54, 8, 96, 58, 23, 31, 76, 41, 18, 130, 108, 146, 117, 5, 39, 95, 72, 79, 147, 3, 74, 46, 99, 114, 77, 10, 30, 35, 148, 134, 102, 135, 65, 40, 82, 107, 129, 106, 144, 17, 16, 26, 103, 51, 25, 119, 32, 101, 98, 71, 126, 112, 47, 100, 85, 37, 88, 36, 45, 83, 78, 29, 92, 38, 113, 11, 21, 61, 136, 34, 142, 143, 127, 120, 59, 50, 43, 53, 20, 80, 64, 24, 0, 28, 62, 110, 15, 104, 56, 12, 1, 49, 86, 60, 145, 27, 66, 116, 149, 94, 52, 132, 14, 84, 42, 91, 139, 22, 118, 93, 44, 131, 125, 105, 121, 63, 115, 69, 137, 87, 128, 6, 122]	33.01	1052.00

De las tablas anteriores se puede ver como los algoritmos GA y ABC son mucho más rápidos que el ACO, y esto puede ser porque estos dos algoritmos generan combinaciones al azar de las ciudades y las van probando hasta encontrar una con la menor distancia lo que hace que el algoritmo utilizado sea mucho menos complejo; por otro lado, el algoritmo ACO es más lento ya que recorre todo los posibles caminos, y en el código esto se lo realiza con el uso de 3 bucles for, por lo que la complejidad es bastante alta; sin embargo, los tiempos de ejecución, a pesar de ser grandes, aún son manejables y capaces de dar resultados útiles. Se puede concluir que los 3 algoritmos dan buenos resultados. Además, para los 3 algoritmos las distancias son

parecidas y no hay una gran variación, aunque este parámetro no permite realizar una buena comparación ya que los valores de las distancias son al azar y las distancias de GA no se crean de la misma manera que las de ABC y ACO. Finalmente, me gustaría mencionar que para la resolución de este tipo de problemas se utilizó chatGPT para inspirar ideas sobre como resolver el TSP con el algoritmo GA y ABC.

Referencias

Bingham, D. (2013). *BEALE FUNCTION*. Obtenido de BEALE FUNCTION:
<https://www.sfu.ca/~ssurjano/beale.html>

Bingham, D. (2013). *RASTRIGIN FUNCTION*. Obtenido de RASTRIGIN FUNCTION:
<https://www.sfu.ca/~ssurjano/rastr.html>

OpenAI. "ChatGPT." Modelo de lenguaje, OpenAI, 2023,
<https://openai.com/research/chatgpt>.

VampboyCreatesEverything (Dirección). (2018). *Ant Colony Optimization Using Python* [Película].