

2. ALGORTIMOS GENÉTICOS

Ejercicio Propuesto:

1. **Ejecute los dos casos de estudio y explique los resultados de ejecución de cada caso de estudio.**

Para el primer caso de estudio, se utiliza un tipo de evaluación: DEFAULT, un criterio de selección individual: DEFAULT y un nuevo tipo de generación de tipo: DEFAULT.

Para el segundo caso de estudio, se utiliza un tipo de evaluación: POR DISTANCIA, un criterio de selección individual: DISTANCIA MINIMA y un nuevo tipo de generación de tipo: DISTANCIA MINIMA.

El primer caso de estudio logra el objetivo después de 982 iteraciones.

El segundo caso de estudio no logra el objetivo después de 1000 iteraciones.

2. **¿Cuál sería una posible explicación para que el caso 2 no finalice como lo hace el caso 1? Revisar el archivo util.py función distance.**

La función de distancia actual calcula la distancia entre dos listas sumando las diferencias entre los elementos correspondientes de ambas listas y luego agregando la diferencia absoluta en longitud entre las listas. Sin embargo, este método puede no ser adecuado para todas las situaciones, ya que simplemente sumar las diferencias no siempre refleja correctamente la relación entre los elementos de las listas.

Por ejemplo, si tenemos dos listas [1, 2, 3] y [4, 5, 6], la distancia según la función actual sería $(4-1) + (5-2) + (6-3) + (3-0) = 18$. Esto no representa adecuadamente la relación entre las listas, ya que podría haber elementos correspondientes que estén muy alejados y otros que estén muy cercanos, lo que no se reflejaría en la distancia calculada.

3. **Realice una correcta implementación para obtener la distancia/diferencia correcta entre dos individuos en el archivo util.py función distance.**

Para abordar este problema, es preferible utilizar una medida de distancia más apropiada, como la distancia euclidiana o la distancia de Manhattan, que tienen en cuenta tanto la magnitud de las diferencias como su dirección.

Después de usar esta implementación, el algoritmo converge en la iteración 378.

```

Generación 375: GA Workshop USFQ - población: 100 - Aptitud: 1.0
Generación 376: GA Workshop USFQ - población: 100 - Aptitud: 1.0
Generación 377: GA Workshop USFQ - población: 100 - Aptitud: 1.0
Objetivo alcanzado:
Generación 378: GA Workshop! USFQ - Aptitud: 0.0

```

4. ¿Sin alterar el parámetro de mutación `mutation_rate`, se puede implementar algo para mejorar la convergencia y que esta sea más rápida? Implemente cualquier mejora que permita una rápida convergencia. Pista: ¿Tal vez elegir de manera diferente los padres? ¿Realizar otro tipo de mutación o cruce?

Implementamos una forma de selección de mecanismos, hay una llamada selección en torneo, en la cual se selecciona de forma aleatoria individuos y se selecciona los mejores de esos subsets, este proceso añade diversidad y evita una convergencia prematura.

```

Generación 203: GA yorksiop! USFQ - población: 100 - Aptitud: 34.0147027033899
Generación 204: GA Worksiop! USFp - población: 100 - Aptitud: 31.016124838541646
Generación 205: GA Worksiop! USFQ - población: 100 - Aptitud: 1.0
Generación 206: GA WoGksiop! USFQ - población: 100 - Aptitud: 43.01162633521314
Generación 207: GA Workomopm USFQ - población: 100 - Aptitud: 76.26925986267338
Objetivo alcanzado:
Generación 208: GA Workshop! USFQ - Aptitud: 0.0

```

Con este cambio, el algoritmo converge en 208 iteraciones.

5. Cree un nuevo caso de estudio 3. Altere el parámetro de mutación `mutation_rate`, ¿ha beneficiado en algo la convergencia? Qué valores son los más adecuados para este parámetro. ¿Qué conclusión se puede obtener de este cambio?

Best Individual Selection Type	Mutation Rate	Iterations
MIN DISTANCE	0.01	378
MIN DISTANCE	0.02	104
MIN DISTANCE	0.03	120
MIN DISTANCE	0.04	133
MIN DISTANCE	0.005	298
MIN DISTANCE	0.001	No converge

El valor mas optimo en este caso vendría a ser un mutation rate de 0.02, ya que, al aumentar este valor, el problema tiende a converger, pero el número de iteraciones es mayor.

No podemos aumentar tanto el mutation rate porque como observamos, el numero de iteraciones aumenta después de aumentar tanto ese valor.

6. Cree un nuevo caso de estudio 4. Altere el tamaño de la población, ¿es beneficioso o no aumentar la población?

Best Individual Selection Type	Poblation Size	Iterations
MIN DISTANCE	100	378
MIN DISTANCE	50	496
MIN DISTANCE	10	No converge
MIN DISTANCE	200	102
MIN DISTANCE	500	51
MIN DISTANCE	1000	42

Podemos observar que aumentar la población ayuda de forma efectiva a disminuir el número de iteraciones.

7. De todo lo aprendido, cree el caso de estudio definitivo (caso de estudio 5) el cual tiene lo mejor de los ítems 4, 5, 6.

Creamos el caso 5, con el tipo de selección individual: NEW (el que creamos para la pregunta 4 (selección en torneo), usamos un valor de mutation rate 0.02 y un tamaño de población 1000.

Observamos que el numero de iteraciones para llegar al objetivo es de 31.