I Datos Generales

Titulo del proyecto:
Travelling salesman problem .

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

Esta función de costo depende de un conjunto de variables que describen las posibles configuraciones, o estados internos, que el sistema puede tomar. En el caso de la optimización combinatoria, el número de estados es finito. Sin embargo, para sistemas grandes a menudo es imposible o prohibitiva mente costoso encontrar la solución exacta. La especificación del circuito más corto que conecta N puntos dados, conocido como el *Traveling Salesman Problem (TSP).* ; es un ejemplo clásico de un problema intratable: la solución, que consiste en un polígono de perímetro mínimo que tiene los puntos N como vértices, requiere operaciones $O(K^N)$. (Bonomi & Lutton, 1984).

Este número aumenta más rápido que cualquier potencia de N. En consecuencia, este problema pertenece a una familia de problemas equivalentes clasificados como NP-hard. Sin embargo, parece probable que ninguno de estos problemas equivalentes pueda resolverse en tiempo polinómico. Sin embargo, debido a su importancia estratégica, se ha hecho un gran esfuerzo para encontrar algoritmos de solución aproximada cuyo tiempo de ejecución sea proporcional a las pequeñas potencias de N.(Bonomi & Lutton, 1984).

En cualquier Travelling salesman problem de N -city las posibles convinaciones esta definido por:

(N-l)! / 2

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 las 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.

Partially-mapped crossover (PMX).
El concepto PMx fue introducido por primera vez por Goldberg y Lingle (1985) que sugirieron un nuevo tipo de operador de crossover, el *"Partially-mapped crossover (PMX)"*, que creen que conducirá a una solución más eficiente a *The travelling salesman problem. PMX* , funciona de la siguiente manera: considere dos posibles codificaciones de un recorrido por ocho ciudades, A_1 y A_2 , un retorno a la ciudad inicial está implícito: (B. Fogel, 1988).

$A_1 = [3, 5, 1, 2, 7, 6, 8, 4]$.

$A_2 = [1, 8, 5, 4, 3, 6, 2, 7]$.

Dos posiciones se determinan de manera aleatoria a lo largo de la codificación de A_1 . Las ciudades reales ubicadas entre estas posiciones a lo largo de A_1 se intercambian con las ciudades entre las mismas posiciones a lo largo de A_2 . Por ejemplo, si se eligen las posiciones tres y cinco, la subcodificación a lo largo de A_1 1-2-7, y la subcodificación a lo largo de A_2 es 5-4-3. Cada una de estas ciudades se intercambia, lo que lleva a los nuevos recorridos, A_1* y A_2* : (B. Fogel, 1988).

$A_1* = [7, 1, 5, 4, 3, 6, 8, 2]$.

$A_2* = [5, 8, 1, 2, 7, 6, 4, 3]$.

The first variant of order crossover (Ox1) .
The first variant of order crossover (Ox1) fue introducido por primera vez por Lawrence Davis en el año 1985, que se construye a partir de elegir una subcodifcación de un recorrido de uno de los padres y preservar el orden relativo de las ciudades del otro padre. Por ejemplo, considere dos padres p_1 y p_2 (con dos puntos marcados con "|") (Deep, Kusum,& Mebrahtu, Hadush (2011).

$P_1 = [1, 2|3, 4, 5|6, 7, 8, 9]$.

$P_2 = [8, 5|7, 1, 2|4, 9, 3, 6]$.

Primero, los segmentos entre los puntos de corte se copian en muelles.

$O_1 = [-, -|3, 4, 5|-, -, -, -]$.

$O_2 = [-, -|7, 1, 2|-, -, -, -]$.

Luego, comenzando desde el segundo punto de corte de uno de los padres, las ciudades del otro padre se copian en el mismo orden, omitiendo las que ya existen. Al llegar al final de la cadena / cromosoma / continuamos desde el primer lugar de la cadena. (Deep, Kusum,& Mebrahtu, Hadush (2011).

La secuencia de ciudades en el segundo padre (desde el segundo punto de corte) es:

$$T_a[4, 9, 3, 6, 8, 5, 7, 1, 2] \ .$$

De esto, después de la eliminación de las ciudades 3, 4 y 5 que ya están en la primera descendencia, obtenemos :

$$T_a[9, 6, 8, 7, 1, 2] \ .$$

Al colocar esta secuencia en la primera descendencia (a partir del segundo punto de corte), tenemos :

$$O_1 = [1, 2|3, 4, 5|9, 6, 8, 7] \ .$$

$$O_2 = [4, 5|7, 1, 2|6, 8, 9, 3] \ .$$

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: Travellng Salesman Problem

input : Random poblation of the size l
output: A approximate solution

```
special treatment of the first line;
for i ← 0 to l do
    for j ← 0 to w do
        pred1,pred2 ← sel_ pad_ best;
        if crossover_ tec equal to 1 then
            desc1,desc2 ← crossover_ PMx_ 18;
        else
            desc1,desc2 ← order_ crossover_ David;
        end
    end
     best generation ← biology_ competition;
    if a is equal to b then
        mutated generation ← Heuristic_ Mutation;
    end
end
```

V.II Diagrama de flujo.

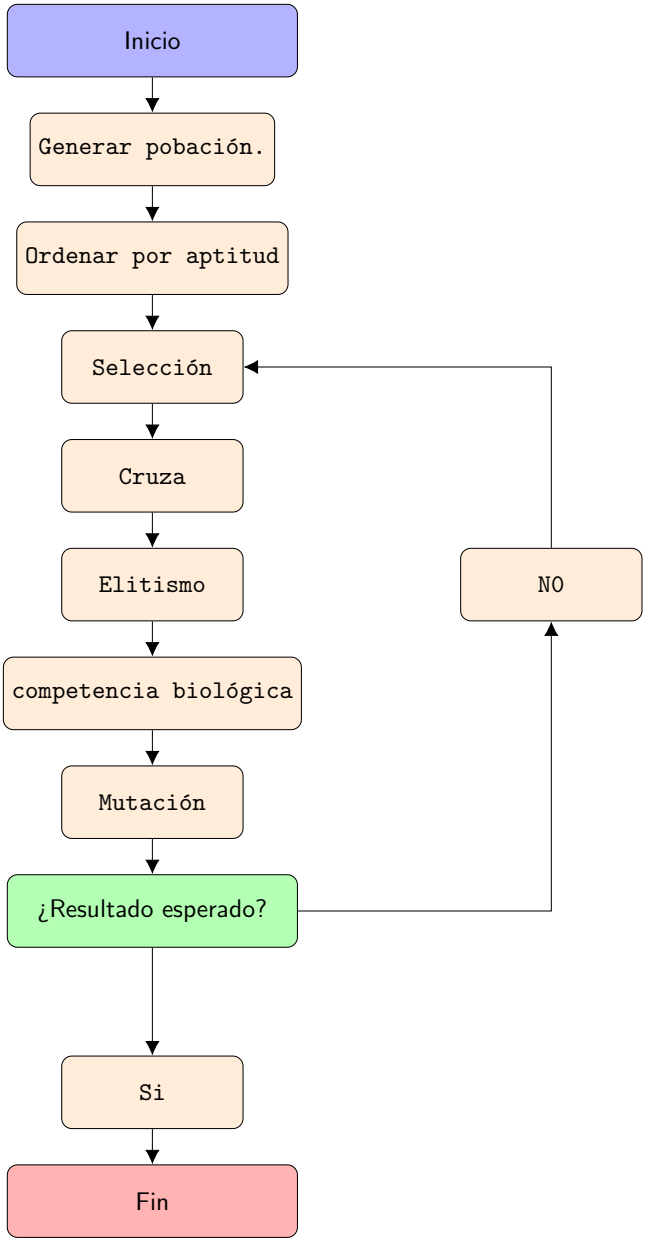

Figure 1: Diagrama de flujo.

VI Desarrollo.

Población.
Para este trabajo en particular se opto por una población de 100 posibles soluciones, estas generadas de manera aleatoria, cada cromosoma, cuenta con 18 genes. Se llama la función *make_ dist_ apt* , dando como resultado una celda con nuestras posibles soluciones, la distancia de cada una, además de la aptitud individual que se considera como la inversa de la distancia ordenados del mas apto al menos.

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* , dentro de esta función se hace uso de la función *my_ own_ RWS_ best* , 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.
Se utiliza la variable *crossover_ tec* , dependiendo de su valor, se manda llamar la técnica *crossoover*, para el valor de uno se llama a la función *crossover_ Pmx_ 18* que es el método de cruza *Partially Mapped Crossover* o cero para llamar a la técnia *order crossover David* , contenida dentro de la función ordder_ crossover_ David.

Partially Mapped Crossover y *Order crossover David* se abordaron con mayor detalle en el sección de métodos.

Competencia Biológica.
La función *biology_ competition* 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* que a su vez manda llamar a la función *scramble_ met_ per_ one*, que como su nombre lo menciona, tiene el propósito de eliminar soluciones repetidas por medio de la técnica de mutación Scramble, evitando así una perdida 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 *make_ dist_ apt*, dando como resultado de la función *biology_ competition* a los individuos con mayor aptitud.

Mutación Heurística.
Después de cada 10 generaciones se manda llamar la función *heuristic_ mutation*, tomando en cuenta el concepto de explotación, se muta el 50 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*, 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 Resutados.

Dentro de las 100 generaciones con una población de 100 posibles soluciones, se obtiene la mejor de una distancia total de 23861.88, con la combinación de ciudades como se muestra a continuación.

A = [7,9,6,1,4,10,15,18,2,5,13,17,16,11,12,14,8,3]

Sustituyendo los números por las posibles soluciones, además de agregar las distancias entre estas, nos encontramos con la siguiente secuencia.

Merida $\Longrightarrow$ 1211.75 $\Longrightarrow$ Monterrey $\Longrightarrow$ 639.86 $\Longrightarrow$ Guadalajara $\Longrightarrow$ 461.58 $\Longrightarrow$ Ciudad de México $\Longrightarrow$ 1235.34 $\Longrightarrow$ San Salvador $\Longrightarrow$ 365.57 $\Longrightarrow$ Managua $\Longrightarrow$ 720.46 $\Longrightarrow$ Panama City $\Longrightarrow$ 861.49 $\Longrightarrow$ Bogota $\Longrightarrow$ 722.17 $\Longrightarrow$ Quito $\Longrightarrow$ 3778.4 $\Longrightarrow$ Mendoza $\Longrightarrow$ 986.54 $\Longrightarrow$ Buenos Aires $\Longrightarrow$ 205.46 $\Longrightarrow$ Montevideo $\Longrightarrow$ 2281.93 $\Longrightarrow$ Brasilia $\Longrightarrow$ 3591.13 $\Longrightarrow$ Caracas $\Longrightarrow$ 3573.96 $\Longrightarrow$ Boston $\Longrightarrow$ 306.28 $\Longrightarrow$ New York $\Longrightarrow$ 357.95 $\Longrightarrow$ Washigton $\Longrightarrow$ 1490.08 $\Longrightarrow$ Miami $\Longrightarrow$ 1101.93 $\Longrightarrow$ Merida.

En la figura 2, se observa el grafo resultante como la mejor aproximación a la solución.

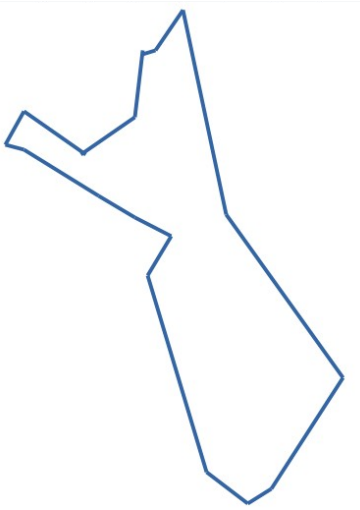

Figure 2: Grafo TSP.

Podemos observar en la figura 3 la aproximación a mejor solución representada sobre el continente Americano.

Figure 3: America TSP.

En la figura 4 se observa la aptitud por generación usando el método de cruza PMx y CX_1

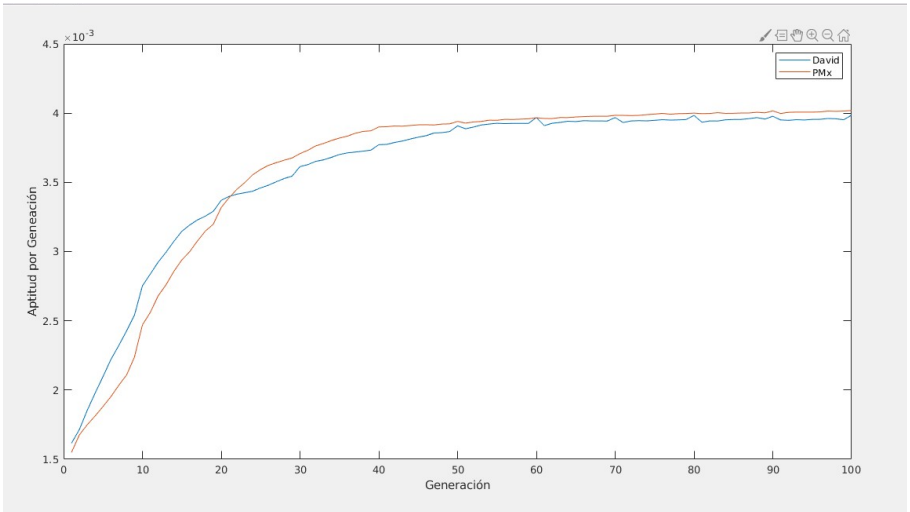

Figure 4: Aptitud por Generación.

En la figura 5 se observa la aptitud del individuo mas apto por generación usando el método de cruza PMx y CX_1

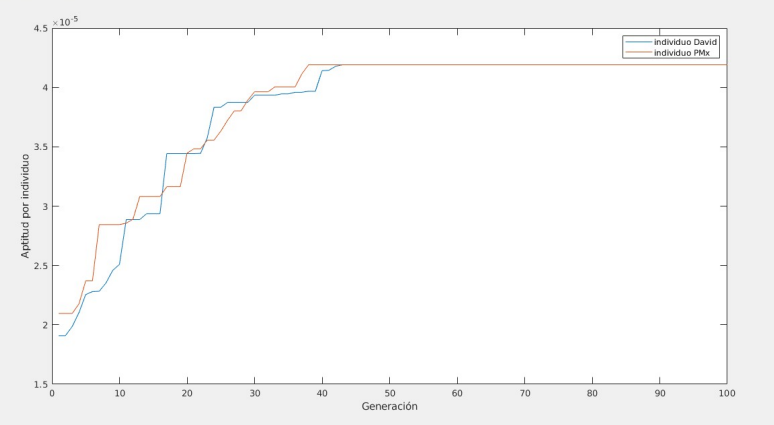

Figure 5: Aptitud por individuo.

VIII Discusión de los Resutados.

Los resultados obtenidos tanto con PMx como con OX_1 , son aproximaciones aceptables, además de no alegarse en capacidad, midiendo esta como el número de generaciones que les cuesta encontrar la mejor aproximación (23861.88), en cambio los dos métodos de mutación, representan dos oportunidades diferentes, que trabajando en conjunto generan buenos resultados.

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.

Para este TSP, podemos decir que el método de mutación heurística nos ayuda a tener una buena explotación y por otro lado el método de mutación *Scramble*, nos ayuda a tener una mayor diversidad y por consiguiente mejor exploración.

X Bibliografía.

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.