

I Datos Generales

Titulo del proyecto:
Travelling Salesman Problem con restricciones .

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

Se han realizado múltiples esfuerzos para aumentar la eficiencia de los Algoritmos Genéticos (AG), y uno de las más prometedoras alternativas es utilizar la implementación paralela. La naturaleza paralela de los algoritmos genéticos ha sido reconocida durante mucho tiempo, y muchos han utilizado con éxito algoritmos genéticos paralelos para reducir el tiempo requerido para alcanzar soluciones aceptables a problemas complejos. Los AG trabajan con una población de soluciones independientes, lo que facilita la distribución de la carga computacional entre varios procesadores. De hecho, los AG son "vergonzosamente paralelos" en el sentido de que es muy fácil implementarlos de manera eficiente en computadoras paralelas. Sin embargo, a pesar de su simplicidad operativa, los GA paralelos son algoritmos no lineales complejos que están controlados por muchos parámetros que afectan su eficiencia y la calidad de su búsqueda. Establecer esos parámetros correctamente es crucial para obtener buenas soluciones de forma rápida y confiable (Cantú-Paz, E. 2012).

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.

III.I *Parallel Models*

Master-Slave Model.

El modelo Master-Slave es fácil de visualizar. Consiste en distribuir las evaluaciones de la función objetivo entre varios procesadores esclavos mientras el bucle principal de el AG se ejecuta en un procesador maestro. Este paradigma paralelo es bastante simple de implementar y su exploración del espacio de búsqueda es conceptualmente idéntico a la de un AG que se ejecuta en un procesador en serie. En otras palabras, el número de procesadores que se utilizan es independiente de las soluciones que se evalúan, excepto por el tiempo. Este paradigma se ilustra en la Figura 1, donde el procesador maestro envía parámetros (los necesarios para la evaluación de la función objetivo) a los esclavos, los valores de la función objetivo se devuelven cuando estos se calculan (Luque, G., & Alba, E. 2011).

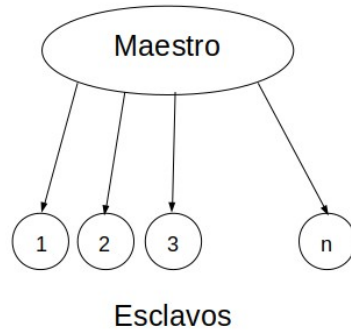


Figure 1: Maestros y esclavos.

El procesador maestro controla la paralelización de las tareas de evaluación de la función objetivo (y posiblemente la asignación y / o transformación del estado físico) realizadas por los esclavos. Este modelo es generalmente más eficiente ya que la evaluación objetiva se vuelve más costosa de calcular, ya que la sobrecarga de comunicación es insignificante con respecto al tiempo de evaluación de aptitud. Sin embargo, tiende a presentar un cuello de botella para algunas docenas de procesadores, lo que generalmente evita la escalabilidad (Luque, G., & Alba, E. 2011).

Distributed Model.

En los AG distribuidos, la población se estructura en sub-poblaciones más pequeñas, relativamente aisladas una de las otras. Los AG paralelos basados en este paradigma a veces se llaman AG multi-población o multi-deme. Independientemente de su nombre, la característica clave de este tipo de algoritmo es que (una copia de) los individuos dentro de una sub-población particular (o isla) ocasionalmente pueden migrar a otra. Este paradigma se ilustra en la figura 2. Tenga en cuenta que los canales de comunicación que se muestran son lógicos, asignaciones específicas se hacen como parte de la estrategia de migración de AG y están asignados a alguna red física (Luque, G., & Alba, E. 2011).

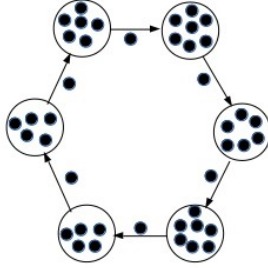


Figure 2: Modelo distribuido.

Conceptualmente, la población general de AG se divide en varias subpoblaciones (o demes) independientes y separadas. Una vista alternativa es la de varios AG pequeños y separados que se ejecutan simultáneamente. Los individuos ocasionalmente migran entre una isla en particular y sus vecinos, aunque estas islas generalmente evolucionan de manera aislada durante la mayor parte del tiempo de ejecución del AG. Aquí, los operadores genéticos (selección, mutación y recombinación) tienen lugar en cada isla, lo que significa que cada isla puede buscar en regiones muy diferentes de todo el espacio de búsqueda con respecto a las demás. Como se dijo antes, cada isla también podría tener diferentes valores de parámetros (Luque, G., & Alba, E. 2011).

El modelo distribuido requiere la identificación de una política de migración adecuada. Los principales parámetros de la política de migración incluyen los siguientes:

Migration Gap.

La brecha migratoria dado que un AG generalmente realiza intercambios dispersos de individuos entre las subpoblaciones, debemos definir la brecha de migración, este es el número de pasos en cada subpoblación entre dos intercambios sucesivos (pasos de evolución aislada). Se puede activar en cada subpoblación, ya sea periódicamente o mediante el uso de una probabilidad dada P_M para decidir en cada paso si la migración se llevará a cabo o no (Luque, G., & Alba, E. 2011).

La migración puede causar una convergencia más rápida. La selección de

migrantes y reemplazos de acuerdo con su aptitud (en lugar de elegirlos al azar) aumenta la velocidad de convergencia. Aunque esto es deseable, debemos ser conscientes de que el algoritmo puede converger demasiado rápido a una solución subóptima (Paz, 2001).

Migration Rate.

La tasa de migración es el parámetro que determina el número de individuos que se someten a migración en cada intercambio. Su valor se puede dar como un porcentaje del tamaño de la población o también como un valor absoluto (Luque, G., & Alba, E. 2011).

Selection/Replacement of Migrants.

La selección y el reemplazo de migrantes es el parámetro que decide cómo seleccionar soluciones para emigrantes y qué soluciones deben ser reemplazadas por los inmigrantes. Es muy común en las AG distribuidas en paralelo usar los mismos operadores de selección / reemplazo para tratar con los migrantes (Luque, G., & Alba, E. 2011).

Best/bad.

Los individuos que migran desde la subpoblación actual se seleccionan entre los mejores individuos, es decir, el que tiene el mayor valor de aptitud física en la subpoblación actual, y reemplazan a los peores individuos con la aptitud física más baja en la próxima subpoblación. (Belkadi, Gourgand, & Mohammed, 2006)

Random.

Los individuos que migran desde la subpoblación actual se seleccionan al azar y reemplazan a los individuos elegidos al azar en la próxima subpoblación. (Belkadi et al., 2006)

Topology.

La Topología es el parámetro que define el vecino de cada isla, es decir, las islas que una subpoblación concreta puede enviar (o recibir de) individuos. La nomenclatura tradicional divide los AG paralelos en modelos de trampolín e islas, dependiendo de si los individuos pueden migrar libremente a cualquier subpoblación o si están restringidos a migrar a islas geográficamente cercanas, respectivamente (Luque, G., & Alba, E. 2011).

Ring topology.

En la topología de anillo, se establece una interconexión lógica o un vecino lógico entre subpoblaciones. Cada una de las subpoblaciones está conectada (lógicamente) a dos subpoblaciones, una como aguas arriba y la otra aguas abajo. Durante una migración, se establece una elección de dirección de migración. Esta dirección determina para cada subpoblación su subpoblación vecina. Por lo tanto, puede enviar y recibir a sus individuos. Esta dirección de migración es respetada por todas las subpoblaciones (Belkadi et al., 2006).

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.

Stochastic Universal Sampling (SUS).

El algoritmo SUS es una variante de RWS destinada a reducir el riesgo de convergencia prematura. Se puede implementar de acuerdo con el siguiente pseudocódigo: (Jebari, 2013)

Algorithm 1: Stochastic Universal Sampling(SUS)

input : Fitness Poblacion
output: j

```
1 Calculate the mean  $\hat{f} = 1/n \sum_{i=1}^n f(i)$  ;  
2 Generate a random number  $\alpha \in [0, 1]$  ;  
3  $sum = f(1)$ ;  $delta = \alpha \times \hat{f}$  ;  $j = 0$ ;  
4 do  
5   if  $delta < sum$  then  
6     select the jth individual ;  
7      $delta = delta + sum$ ;  
8   else  
9      $j = j + 1$ ;;  
10     $sum = sum + f(j)$ ;  
11  end  
12 while  $j < n$ ;
```

Linear Rank Selection (LRS)

. LRS también es una variante de RWS que intenta superar el inconveniente de la convergencia prematura de la AG a un óptimo local. Se basa en el rango de las personas y no en su aptitud. El rango n se otorga al mejor individuo, mientras que el peor individuo obtiene el rango 1. Por lo tanto, en función de su rango, cada individuo i tiene la probabilidad de ser seleccionado dada por la expresión: (Jebari, 2013)

$$p(i) = \frac{rank(i)}{n(n-1)} \quad (1)$$

Una vez que todos los individuos de la población actual están clasificados, el procedimiento LRS puede implementarse de acuerdo con el siguiente pseudocódigo:

Algorithm 2: Linear Rank Selection(LRS)

input : Poblacion
output: index

```
1 Calculate the sum  $v = \frac{1}{n-2.001}$  ;  
2 For each individual  $1 \leq j \leq n$  ;  
3 do  
4   Generate a random number  $\alpha \in [0, v]$  ;  
5   if  $p(j) \leq \alpha$  then  
6     select the  $j$ th individual ;  
7   end  
8 while;
```

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 subcodificación 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 :

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

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

Complete Subtour Exchange Crossover (CSEX).

El operador completo de cruce de intercambio de subtour (CSEX) es diseñado para operar con la representación de ruta. CSEX fue propuesto por Katayama y Narihisa para ser utilizado específicamente para problemas de permutación (como el TSP). La filosofía detrás de CSEX es alentar a la descendencia a heredar tantos buenos rasgos (sustancia) de los padres como sea posible. CSEX enumera sustancias que tienen la misma dirección (o dirección invertidas) en dos permutaciones como sustancia común (Umbarkar & Sheth, 2015)

$$Parent1 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] .$$

$$Parent2 = [4, 9, 7, 6, 5, 0, 8, 2, 1, 3] .$$

$$Offspring1 = [0, 2, 1, 3, 4, 5, 6, 7, 8, 9] .$$

$$Offspring2 = [0, 1, 2, 3, 4, 7, 6, 5, 8, 9] .$$

$$Offspring3 = [0, 2, 1, 3, 4, 7, 6, 5, 8, 9] .$$

$$Offspring4 = [4, 9, 5, 6, 7, 0, 8, 2, 1, 3] .$$

$$Offspring5 = [4, 9, 7, 6, 5, 0, 8, 1, 2, 3] .$$

$$Offspring6 = [4, 9, 5, 6, 7, 0, 8, 1, 2, 3] .$$

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 3: Travelling Salesman Problem with restriction

input : Random poblacion of the size l

output: A approximate solution

1 *Generations* $\rightarrow 500$;

2 **for** $0 \rightarrow \text{Generations}$ **do**

3 $\text{migration, index} \leftarrow \text{Migration grap}$;

4 $\text{fitness, poblacion dolphin, migrator} \leftarrow \text{Society}$
 Dolphin(migrator) ;

5 $\text{fitness, poblacion fox, migrator} \leftarrow \text{Society Fox(migrator)}$;

6 $\text{fitness, poblacion Raven, migrator} \leftarrow \text{Society Raven(migrator)}$;

7 **end**

V.II Diagrama de flujo.

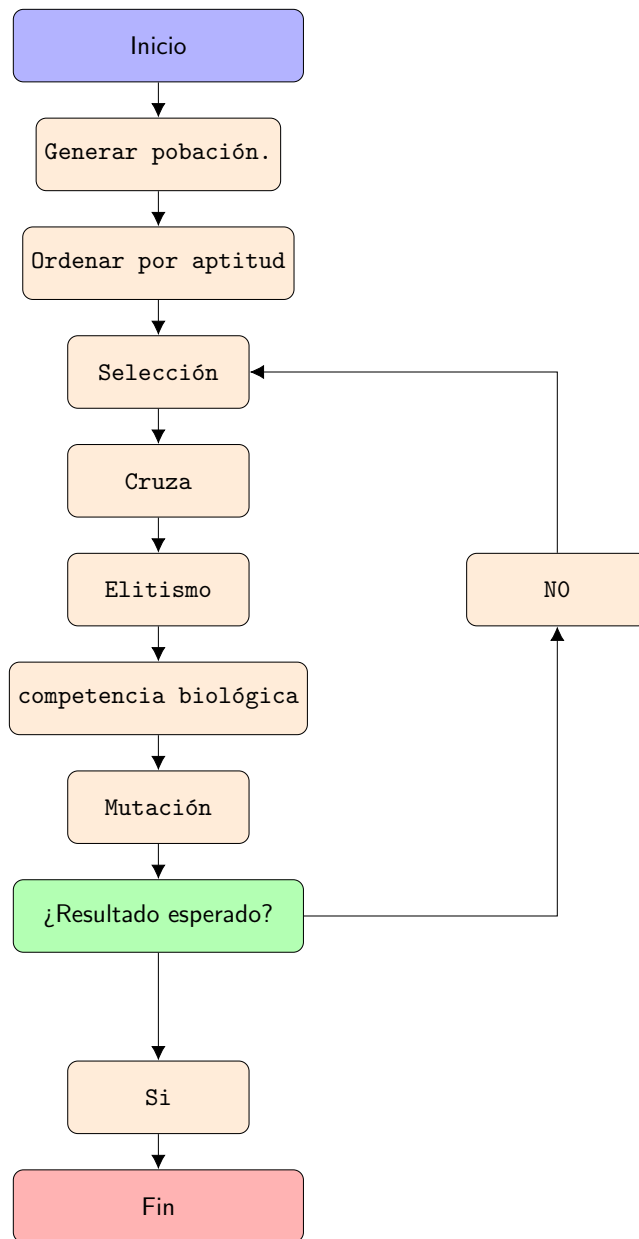


Figure 3: Diagrama de flujo.

VI Desarrollo.

La topología seleccionada para el trabajo fue de tipo anillo, con tres sociedades diferentes llamadas Dolphin, Fox, Raven, comunicadas solo con sus vecinos más próximos, el sentido de la migración fue en sentido a las manecillas del reloj, se observa en la figura 4, la representación gráfica.

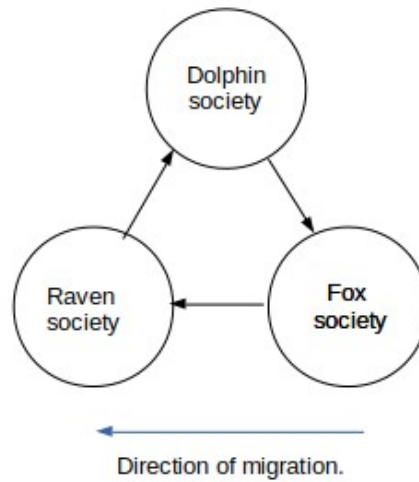


Figure 4: Ring topology.

Dentro de los parámetros de migración, se mencionan, la migración aleatoria y la migración mejor – peor, como su nombre describe la primera consiste en seleccionar un individuo aleatoria mente y este migrar a la siguiente sociedad, ocupando un lugar aleatorio del individuo que dejo esa sociedad, generando así un proceso de migración para las tres sociedades, en la segunda técnica, se manda el clon de el mejor individuo a migrar a la siguiente sociedad, remplazando al individuo menos apto de esta, al igual en la sociedad que dejo, llegara el clon del individuo más apto de la sociedad anterior, esto en sentido del reloj, la frecuencia con la que se genera la migración depende en gran medida de a técnica, hablamos cerca de cada 15 generaciones un individuo aleatorio y 30 un clon del mejor individuo, en la sección de resul-

tados, se muestra la comparación de las diferentes técnicas de migración.

En la figura 5, se muestra el diagrama de un conjunto de sociedades denominado universo, cada sociedad es diferenciada una de la otra por sus técnicas de selección y de cruce entre individuos, la sociedad Dolphin, se compone por el método de selección Rolette Wheel selection y Partially Mapped Crossover como técnica de cruce, la sociedad Fox tiene Stochastic Universal Sampling como técnica de selección y First Variant of Order crossover como técnica de cruce, la ultima sociedad, Raven, usa Linear Rank Selection como técnica de selección y Compete Subtour Exchange Crossover como técnica de cruce.

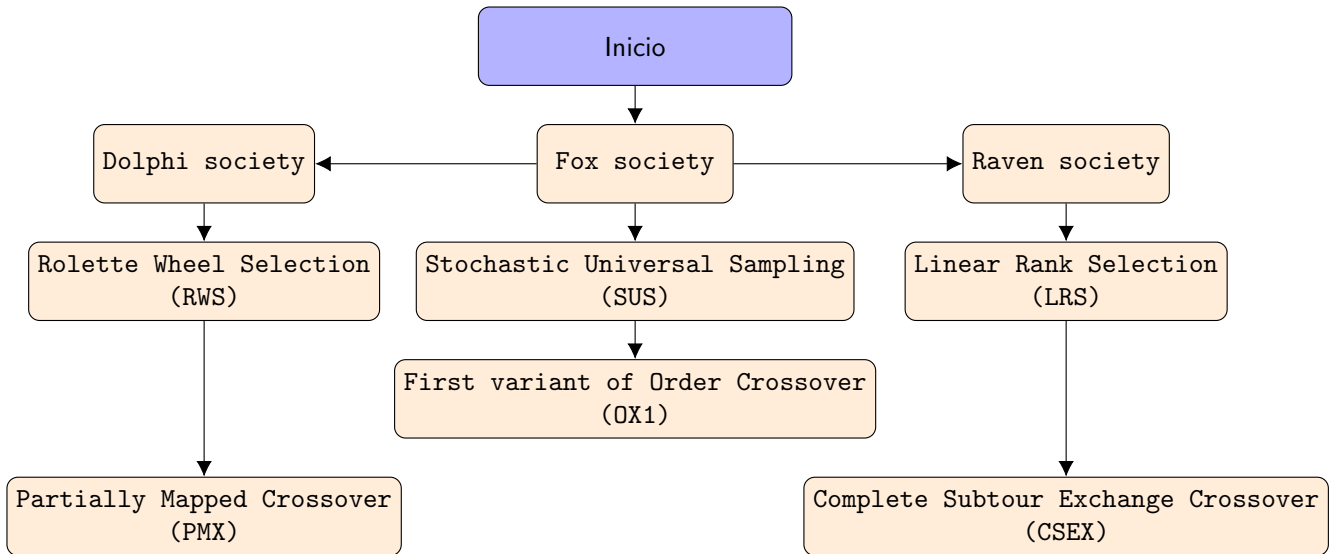


Figure 5: Diagrama de flujo sociedades.

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 or-

denados del mas apto al menos.

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 Resultados.

Los resultados obtenidos después de las 550 generaciones se muestran en la figura 6, esta bajo las condiciones de migración de cada 10 generaciones, un individuo seleccionado aleatoria mente, migrara a la siguiente sociedad.

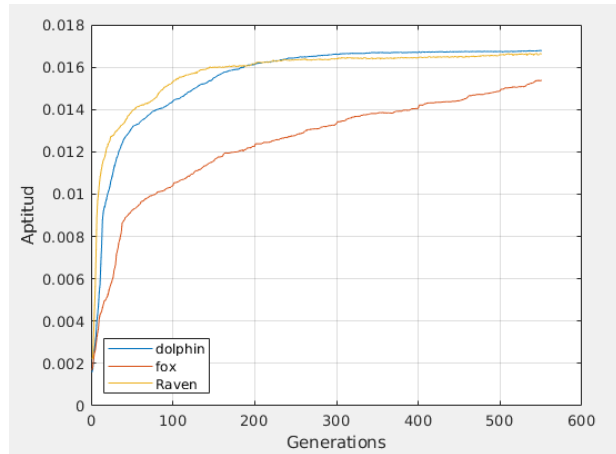


Figure 6: Resultados de migración aleatoria.

En la figura 7 se muestra el comportamiento de las sociedades bajo las condiciones de migración de best/worst, es decir tomar el mejor en aptitud de una generación, y mandar una copia a la siguiente generación, ubicándolo en el lugar del peor individuo en aptitud.

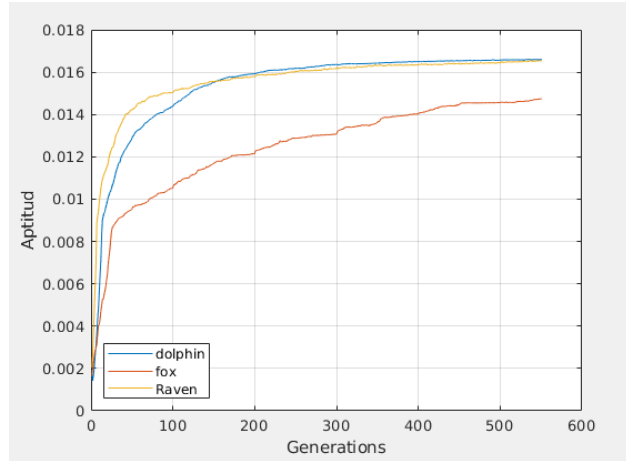


Figure 7: Resultados de migración best/worst.

En la figura 8, se muestra el crecimiento de las sociedades, sin migración, desarrollándose solamente con sus propios métodos he individuos.

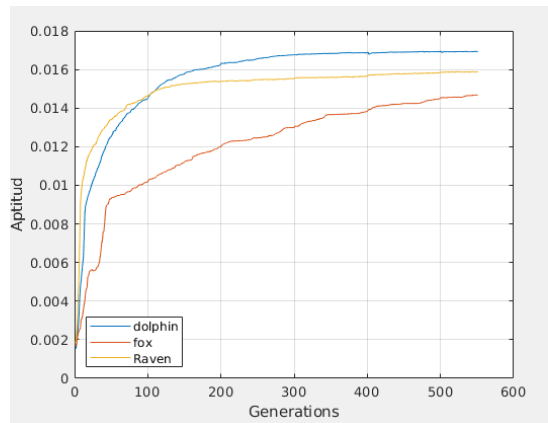


Figure 8: Resultados sin migración.

En la figura 9 se muestra el comportamiento de las diferentes sociedades, utilizando ambos métodos de migración, además, de aumentar la constan-

cia de esta, siendo cada 5 generaciones la migración random, y cada 30 la migración Best/Worst.

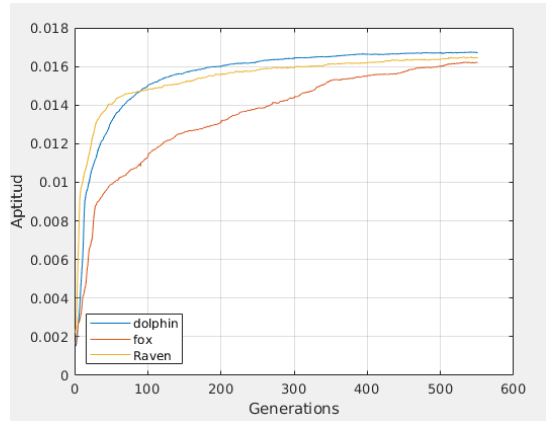


Figure 9: Resultados con ambas migración.

En la tabla 1, se muestra los mejores soluciones generadas por las diferentes soluciones, con migración random y Best/Worst 5 y 30 generaciones respectivamente, y en la figura 10 el despeno en aptitud.

Society	Solution	Distance	Fitness	migration
Dolphin	16,18,17,12,3,4,15,14 ,1,2,9,10,11,5,7,8,13,6	5691	1.7572e-04	random- Best/Worst
Fox	16,18,17,12,3,4,15,14 ,1,2,9,10,11,5,7,8,13,6	5691	1.7572e-04	random- Best/Worst
Raven	16,18,17,12,3,4,15,14 ,1,2,9,10,11,5,7,8,13,6	5691	1.7572e-04	random- Best/Worst
Dolphin	9,2,1,14,15,4,3,12,17 ,18,16,6,13,8,7,5,11,10	5691	1.7572e-04	random
Fox	9,2,1,14,15,4,3,12,18 ,16,17,13,6,8,7,5,11,10	5722	1.7476e-04	random
Raven	10,11,9,2,1,14,15,4,3 ,12,13,6,16,18,17,8,7,5	5913	1.6912e-04	random
Dolphin	9,10,11,5,7,8,6,13,12 ,3,4,17,16,18,14,15,1,2	5673	1.7627e-04	Best/Worst
Fox	9,10,11,5,7,8,6,13,12 ,3,4,17,16,18,14,15,1,2	5673	1.7627e-04	Best/Worst
Raven	9,10,11,5,7,8,6,13,12 ,3,4,17,16,18,14,15,1,2	5673	1.7627e-04	Best/Worst
Dolphi	17,16,18,14,15,1,2,9,10 ,11,5,7,8,6,13,12,3,4	5673	1.7627e-04	without mi- gration
Fox	11,3,12,6,7,8,13,17,16 ,18,14,4,15,1,2,9,5,10	6288	1.5903e-04	without mi- gration
Raven	2,9,10,11,5,7,8,6,13 ,12,17,16,18,14,15,4,3,1	5718	1.7489e-04	wihout migration

Table 1: Tabla resultados por sociedad .

En la figura 11 se muestra el recorrido final, sobre las respectivas ciudades.

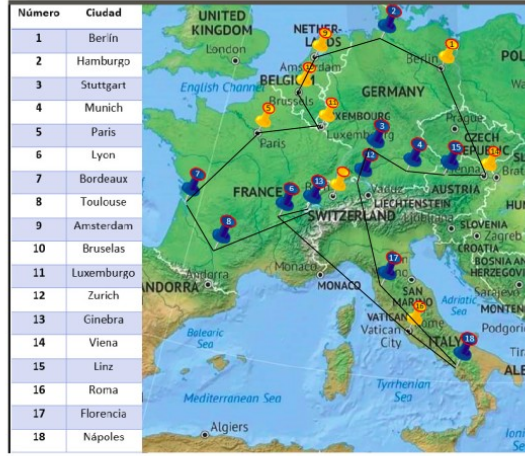


Figure 10: Posible ruta final.

VIII Discusión de los Resultados.

En la sección de resultados , en la figura 7 se observa que el desplazamiento de la aptitud de la sociedad Fox, es la que más le cuesta encontrar la solución, si miramos dentro de la sociedad, nos encontramos con su método de selección *Stochastic Universal Sampling*, esta técnica, usualmente da como resultado (index) valores muy cercanos a 60 -70, sabemos que la exploración es fundamental para el correcto desempeño de un algoritmo genético, entonces se le podría atribuir su dificultad de encontrar la solución, a su método de selección, que de cierta manera, dificulta la exploración, su método de cruza es *First Variant of Order Crossover*, que es un método que fomenta la exploración, pero esta combinación se encuentra en desventaja frente a las demás sociedades, sería conveniente en trabajos futuros pensar en compensar esa falta de exploración.

Por otro lado la sociedad Dolphin muestra la mayor capacidad para encontrar la mejor solución en menos tiempo, tiene como técnica de selección *Roulette Wheel Selection* y *Partially Mapped Crossover* como cruza, este ultimo, en cantidades grandes de individuos, se puede empezar a considerar como un método aleatorio, que ayuda en una mayor exploración, por otro

lado la técnica PMC, en ocasiones, algunos de sus hijos se vuelven una copia exacta de los padres, generando una desventaja, es aquí cuando el método de mutación Escramble es aplicada para remplazar los individuos repetidos, compensando las deficiencias que PMC pudiera presentar.

Entonces, ¿De qué manera afecta la migración a las sociedades?, tenemos dos técnicas de migración, una de manera aleatoria y mejor-peor, en la tabla 1, se observa con mayor claridad las implicaciones de la migración. En el modelo donde se aplican ambas técnicas, todas las sociedades llegan al resultado en las 500 generaciones, si lo analizamos, tomando en cuenta la figura 9, nos damos cuenta que la primera sociedad que encuentra la posible solución (En este caso Dolphin), deberá mandar un clon de su mejor individuo a las demás sociedades, aunque estas no encuentren la solución por si mismas, jamás se verán en la necesidad de abundar al clon enviado por las mejor sociedad, dando como resultado final, el mismo individuo.

El segundo método es seleccionar el individuo a migrar de manera aleatoria, en esta técnica, se observa en la tabla 1 y en la figura 6, a dos sociedades, que llegan a un nivel de aptitud considerable en poco tiempo, pero la tercera sociedad que parece quedarse rezagada, muestra un avance más pronunciado que en la técnica de peor-mejor, entonces, podríamos decir que la migración aleatoria ayuda en la exploración de diferentes soluciones, que por si sola, la sociedad no puede encontrar efectivamente, por otro lado, las dos generaciones más aptas no se ven considerablemente afectadas.

El tercer método que es solo la aplicación de la técnica mejor-peor, tres sociedades llegan a la misma solución, como se comento anteriormente, esto tiene sentido, pero se vuelve notable en la figura 7, como llega un momento que la sociedad menos apta llega a una aparente estagnación, entendemos, que la sociedad por si sola no cuenta con una buena exploración, y esta técnica de migración, definitivamente le dará el resultado, pero no le ayuda a encontrarla por si misma.

En cuarto método, dejando a las sociedades sin migración, se es más notoria la diferencia entre las sociedades, en el aumento de su aptitud general, en su avance en las generaciones, para encontrar un posible solución, se observa a la mejor solución (Dolphin), y la más rezagada (Fox).

IX Conclusión.

Para este trabajo en específico, tomando en cuenta los diferentes métodos implementados, y la manera en que se implementaron, reforzamos el argumento mencionado a lo largo de este trabajo donde hace referencia a la importancia de la exploración para el correcto desempeño de un algoritmo genético, el paralelismo, además de la velocidad del mismo, la migración nos da una herramienta para obtener espacios de búsqueda mayores, como opinión personal, me parece interesante el concepto que pudiera definir a la migración como apoyo a sociedades rezagadas.

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.

Luque, G., & Alba, E. (2011). *Parallel Genetic Algorithms*. New York, Estados Unidos: Springer Publishing. <https://doi.org/10.1007/978-3-642-22084-5>

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

hello we will (Jebari, 2013) and (Paz, 2001)

References

- Belkadi, K., Gourgand, M., & Mohammed, B. (2006, 11). Parallel genetic algorithms with migration for the hybrid flow shop scheduling problem. *JAMDS*, 2006. doi: 10.1155/JAMDS/2006/65746
- Jebari, K. (2013, 12). Selection methods for genetic algorithms. *International Journal of Emerging Sciences*, 3, 333-344.
- Paz, E. C. (2001). *Efficient and accurate parallel genetic algorithms*. Springer Science Business media.
- Umbarkar, D. A., & Sheth, P. (2015, 10). Crossover operators in genetic algorithms: A review. *ICTACT Journal on Soft Computing (Volume: 6 , Issue: 1)*, 6. doi: 10.21917/ijsc.2015.0150