

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO



FACULTAD DE INGENIERÍA MAESTRÍA EN CIENCIAS INTELIGENCIA ARTIFICIAL

Algoritmo Genético Job Shop Scheduling.

Nombre del estudiante:

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.

Tabla de Contenido.

1	Introducción	3
2	Marco Teórico.	4
3	Materiales y Métodos.	8
3.1	Materiales.	8
3.2	Métodos.	8
4	Pseudocódigo y Diagrama de flujo	12
4.1	Pseudocódigo.	12
4.2	Diagrama de flujo.	13
5	Desarrollo.	14
5.1	Arquitectura.	14
5.2	Población.	15
5.3	Evaluación aptitud.	15
5.4	Selección	16
5.5	Cruza.	16
5.6	Competencia Biológica.	16
6	Resutados.	18
6.1	Aptitud Makespan mayor.	18
6.2	Aptitud General.	21
7	Discusión de los Resutados.	23
8	Conclusión.	24
9	Bibliografía.	25
	References	26

Lista de Figuras.

1	Diagrama de Gantt.	8
2	Diagrama de flujo.	13

3	Diagrama	14
4	Diagrama Master-Slave	14
5	Posibles soluciones.	18
6	Digrama de Gantt de posibles soluciones	19
7	Resultados después de 20 generaciones.	20
8	Frecuencia por generación.	20
9	Posibles soluciones.	21
10	Posible solución Diagrama de Gantt.	21
11	Aptitud por generación.	22
12	Frecuencia por generación.	22

Lista de Tablas.

1 Introducción

Dentro de la gran variedad de problemas de programación de producción, el problema general de taller de trabajo (job shop problem por sus siglas en ingles JSP) es probablemente el más estudiado por la investigación académica durante la década de los 90. Se ha ganado la reputación de ser notoriamente difícil de resolver. Ilustra al menos algunas de las demandas requeridas por una amplia gama de problemas del mundo real.(Mattfeld, 1996)

2 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.

3 Materiales y Métodos.

3.1 Materiales.

El algoritmo se programa usando Matlab.

3.2 Métodos.

Población.

La población se genero con cromosomas de 19 genes, cada uno representa una operación diferente de las ocho tareas que se presentan en nuestro trabajo como se observa en el siguiente cromosoma.

La codificación fue dependiendo de el numero de operación O_{ij} , donde i es la operación y j es el número de operación.

Cromosoma = [2 7 3 6 2 2 3 2 6 3 2 4 7 5 1 7 1 3 8];

Para la evaluación de la aptitud se uso la técnica del Diagrama de Gantt, como se observa en la figura 1.

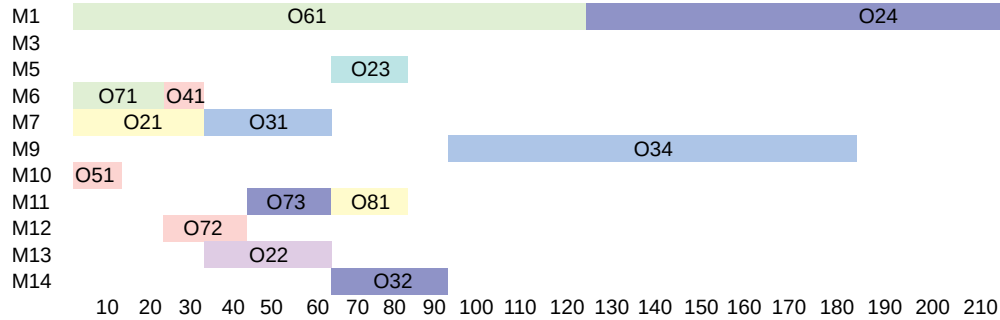


Figura 1: Diagrama de Gantt.

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 1: 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**;

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.

$$A_1 = [1, 4, 6, 3, 5, 7, 8] \text{ .}$$

4 Pseudocódigo y Diagrama de flujo

4.1 Pseudocódigo.

Algorithm 2: Job Shop Scheduling

input : Random poblacion of the size l

output: A approximate solution

1 *Generations* $\rightarrow n$;

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

3 *Possible Soution* \leftarrow *Linear Rank Selection*;

4 *Descendent* \leftarrow *Complete Subtour Exchange Crossover.*;

5 *fitness* \leftarrow *Evaluation* ;

6 *Best Solution* \leftarrow *Biology competition* ;

7 **end**

4.2 Diagrama de flujo.

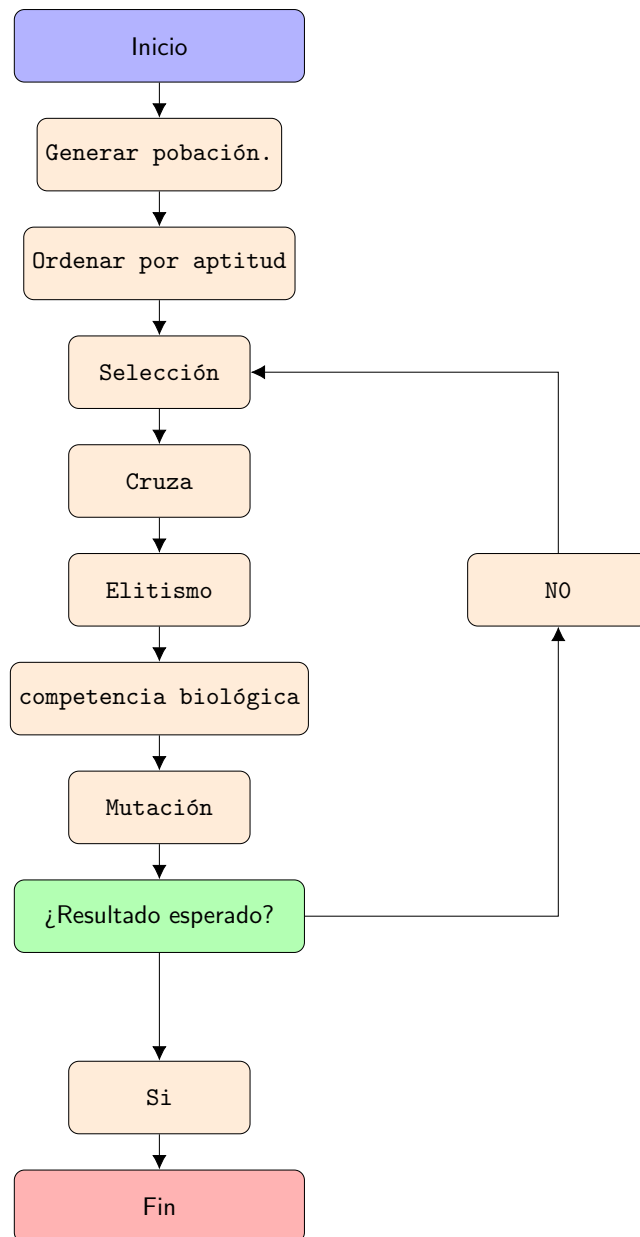


Figura 2: Diagrama de flujo.

5 Desarrollo.

5.1 Arquitectura.

El diagrama de la estructura del algoritmo, se muestra en la figura 3, donde como se mencionara más adelante, se genera una población de cromosomas de 19x1, después para evaluar la aptitud de los mismos, se aplica el diagrama de Gantt para evaluar el Makespan de las diferentes posibles soluciones, para la selección se emplea la técnica LRS y para la crucea CESX.

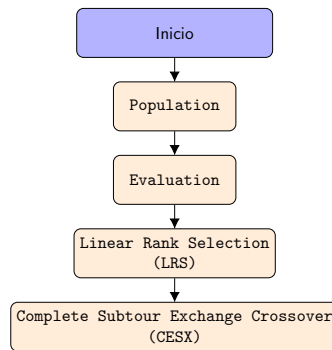


Figura 3: Diagrama

Si consideramos toda la estructura que compone al algoritmo genético, y lo consideramos como una función esclavo, lo metemos dentro de otra a la cual denominamos función maestra, entonces tenemos una arquitectura como la que se muestra en la figura 4.(Regresa, Aptitud General, mejor solución).

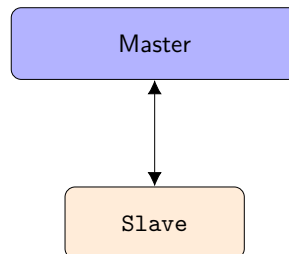


Figura 4: Diagrama Master-Slave

5.2 Población.

Para la población, la codificación fue dependiendo de el numero de operación O_{ij} , donde i es el trabajo y j es la operación. Se puede observar que algunos genes del cromosoma se repiten, esto se debe a que a cada número repetido se le asignara una operación deferente del mismo trabajo, este se asigna en orden de izquierda a derecha, como se muestra en el cromosoma de operaciones, en el ultimo cromosoma, el de máquina, se muestra la máquina donde se ejecutara la operación.

$$\text{Cromosoma} = [3 \ 4 \ 5 \ 1 \ 2 \ 6 \ 7 \ 8 \ 7 \ 3 \ 2 \ 1 \ 6 \ 7 \ 2 \ 3 \ 3 \ 2 \ 2];$$

$$\text{Operación} = [O_{31}O_{41}O_{51}O_{11}O_{21}O_{61}O_{71}O_{81}O_{72}O_{32}O_{22}O_{12}O_{62}O_{73}O_{23}O_{33}O_{34}O_{24}O_{25}];$$

$$\text{Máquina} = [7 \ 6 \ 10 \ 1 \ 7 \ 1 \ 6 \ 11 \ 12 \ 14 \ 13 \ 3 \ 3 \ 11 \ 5 \ 5 \ 9 \ 1 \ 3];$$

Se observa que el cromosoma esta compuesto por 19 genes, esto se debe a que es el número total de operaciones de todos los diferentes trabajos, además se puede observar que las máquinas 2, 4 y 8, no se encuentran en el cromosoma debido a que en ellas no se ejecuta ninguna operación.

5.3 Evaluación aptitud.

Dentro de la evaluación de la aptitud de cada posible solución, se aplicaron dos diferentes acercamientos al problema, uno donde solo se considera la máquina con el mayor tiempo o makespan, un segundo donde se toma en cuenta el tiempo y/o makespan general.

El primero donde la aptitud viene dada por el makespan mayor y se puede representar como:

$$f_i = \max\{M_1, \dots, M_n\} = \begin{cases} M_n & \text{if } M_n > M_{n-1} \\ M_{n-1} & \text{otherwise} \end{cases} \quad (2)$$

Donde f_i representa la aptitud de esa posible solución en cada generación M_n representa el makespan de cada máquina y n el número de máquina.

El segundo acercamiento viene dado por la sumatoria de los tiempo y/o makespan de todas las máquinas por generación, buscando disminuir el tiempo en cada máquina, este se puede representar de la siguiente manera:

$$f_i = (\sum_{n=1}^j (M1, .., Mn))(1/n) \quad (3)$$

$$\hat{f}_i = 1/f_i \quad (4)$$

Donde \hat{f}_i final en cada generación es la inversa de la aptitud por generación f_i , esta es la sumatoria de el tiempo y/o makespan de cada máquina durante esa generación dividido entre el número de máquinas.

5.4 Selección

Para la selección de individuos se uso la técnica de Linear Rank Selection (LRS), como se puede observar en la sección de marco teórico, esta es una variación de Roulette Wheel Selection (RWS).

5.5 Cruza.

Para el método de cruza se utilizo Complete Subtour Exchange Crossover (CESX), esta técnica es diseñada especialmente para problemas como el viajero frecuente o el problema que se aborda en este trabajo.

5.6 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_ap*, dando como resultado de la función *biology_competition* a los individuos con mayor aptitud.

6 Resultados.

6.1 Aptitud Makespan mayor.

Como se observa en figura 5, La naturaleza del problema, tomando en cuenta que buscamos disminuir el makespan, nos encontramos con que el área de posibles soluciones es bastante amplio, en la figura 5 se muestran un conjunto de diferentes posibles soluciones, donde todas ellas generan un makespan = 280.

2	2	7	8	5	2	3	7	6	3	7	3	4	6	2	3	2	1	1
2	2	6	3	7	2	3	2	6	8	2	1	5	7	4	3	1	7	3
6	8	6	2	2	2	3	3	3	4	5	2	7	7	7	3	1	2	1
7	2	4	8	2	3	7	2	6	2	3	7	3	6	1	2	5	1	3
3	2	6	2	2	2	8	7	6	3	2	7	1	4	3	7	1	5	3
2	6	8	3	2	2	2	4	7	3	5	6	7	2	1	7	3	3	1
7	2	2	6	3	8	7	2	3	3	6	7	2	4	3	1	5	2	1
1	8	7	4	3	2	3	2	2	7	7	1	3	6	5	2	6	2	3
7	8	7	6	2	2	2	7	6	2	3	2	5	1	1	3	3	3	4
2	8	2	6	2	4	2	3	7	3	1	5	3	7	6	7	2	3	1
2	7	3	2	7	2	3	6	7	2	5	3	6	1	4	8	3	2	1
3	2	3	5	7	2	6	3	7	2	4	2	1	6	7	2	3	8	1
2	2	2	7	4	3	8	6	2	6	7	1	2	1	3	3	7	5	3
2	2	3	6	3	2	6	5	8	3	2	7	1	4	7	7	3	2	1
6	3	7	7	6	2	3	3	4	7	8	5	2	2	3	2	1	2	1
2	2	2	6	7	7	7	2	3	4	3	5	6	3	1	3	8	2	1
3	2	7	6	2	6	3	5	7	4	2	2	8	1	7	2	3	1	3
5	7	3	6	2	3	6	2	2	7	8	3	2	1	2	1	3	4	7
4	3	2	6	7	5	3	3	8	7	2	3	2	2	1	7	6	2	1
3	8	2	2	7	2	7	6	2	3	3	7	6	5	4	3	2	1	1
3	6	5	3	4	3	6	7	2	8	7	2	7	2	2	1	2	1	3
6	2	2	3	4	6	7	8	7	3	2	2	5	1	3	2	7	1	3
4	3	7	3	7	2	2	5	7	3	8	6	2	2	3	6	2	1	1
3	6	2	7	4	3	2	2	2	7	1	3	8	3	7	6	2	5	1
3	2	2	3	3	2	6	3	6	4	2	7	1	5	8	7	2	7	1
2	4	3	6	7	1	5	3	7	3	6	2	2	2	3	8	7	1	2
7	3	3	3	7	7	2	3	2	6	2	8	2	6	1	4	2	1	5
3	4	2	3	7	7	8	2	3	7	2	6	2	6	5	1	3	2	1
7	5	3	2	3	2	2	6	2	1	6	7	2	8	7	3	3	4	1
3	2	3	2	6	2	3	7	8	4	5	7	7	2	3	1	6	2	1
6	2	2	3	5	6	7	8	2	7	2	3	3	3	7	2	1	4	1
6	2	7	3	2	2	3	8	2	3	7	6	5	1	7	3	2	1	4
2	6	3	3	2	2	2	7	1	3	7	6	5	7	3	4	8	2	1
7	5	3	3	1	3	3	1	4	7	2	7	2	2	6	6	2	8	2
2	3	3	2	2	3	7	6	2	7	7	5	8	6	4	2	1	3	1
8	2	6	3	2	5	3	7	6	4	2	2	1	7	3	2	1	7	3
6	6	3	3	7	2	7	2	2	7	3	2	1	5	8	4	2	3	1
2	2	8	3	5	2	6	6	3	2	3	3	1	2	1	4	7	7	7
6	7	3	6	2	7	3	3	7	2	2	2	4	5	3	8	1	2	1
4	3	7	2	3	2	6	2	6	3	8	5	2	3	2	7	1	1	7
2	7	4	8	3	2	3	7	6	2	6	2	3	2	1	7	1	3	5
2	7	3	2	2	6	2	1	7	3	6	2	3	5	8	1	3	4	7
8	6	7	7	2	2	3	2	2	6	3	1	2	3	4	3	5	1	7
7	2	6	2	8	2	3	6	2	3	7	7	4	1	3	2	1	3	5
2	5	7	6	2	7	8	4	2	3	2	7	1	3	6	3	2	1	3
2	6	6	3	2	2	3	2	3	1	8	4	7	2	7	5	3	1	7

Figura 5: Posibles soluciones.

Aunque existen un gran conjunto de diferentes cromosomas que son la representación de diferentes tipos de soluciones, a la hora de generar su diagrama de Gantt, nos encontramos con que estos son bastante similares, como se muestra en la figura 6

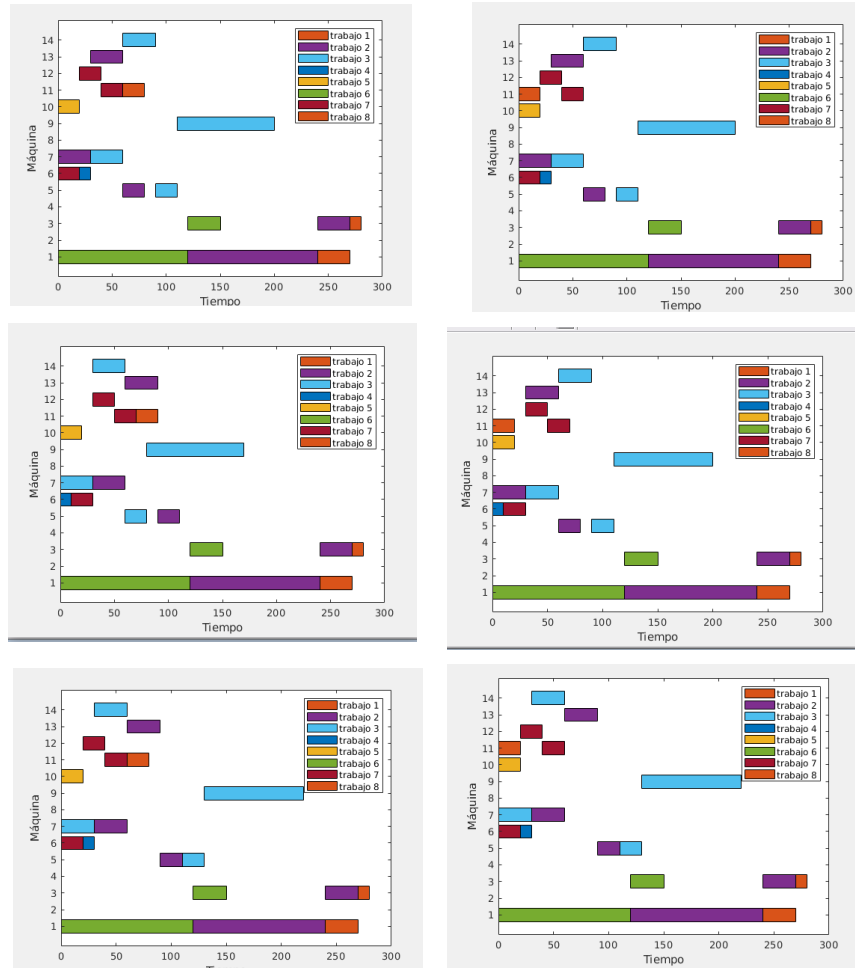


Figura 6: Digrama de Gantt de posibles soluciones

En la figura 7, se observa la grafica de la mejor solución dependiendo de las generaciones, se observa como después de las 20 soluciones encuentra una posible solución

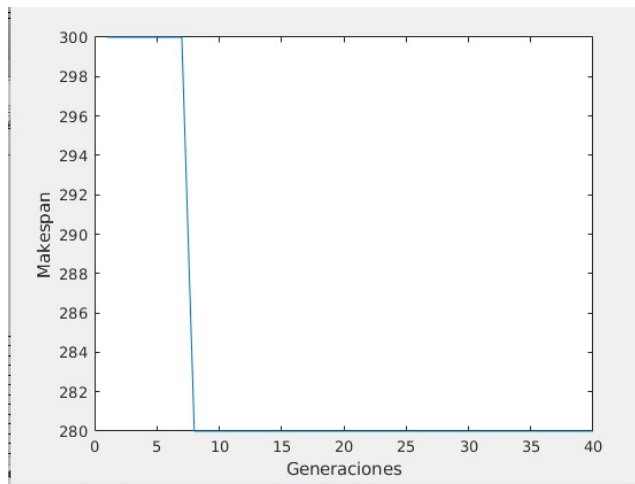


Figura 7: Resultados después de 20 generaciones.

En la figura 8, se muestra la frecuencia con la que encuentra una posible solución en una determinada generación

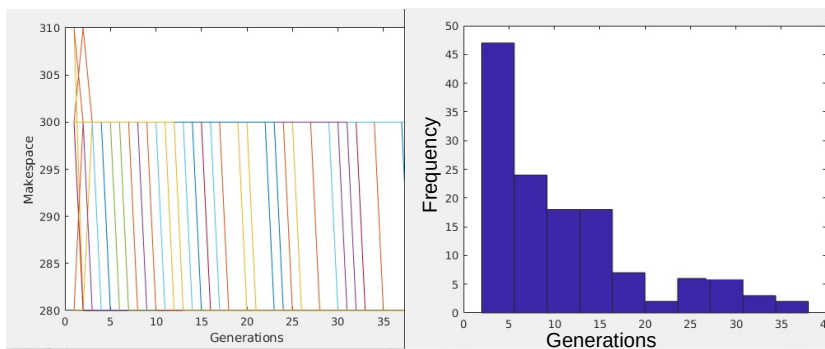


Figura 8: Frecuencia por generación.

6.2 Aptitud General.

El número de posibles soluciones, sigue presentando una gran cantidad de variantes como las que se muestran en la figura 9

7	3	4	6	2	7	3	3	6	2	2	8	5	2	2	3	1	1	7
3	3	7	5	6	8	2	6	2	3	7	2	7	2	2	4	1	1	3
5	7	3	4	2	3	7	6	3	6	2	2	2	3	1	8	2	1	7
3	6	8	5	7	3	2	3	2	2	2	1	4	6	2	1	7	3	7
3	7	4	6	5	7	3	2	8	3	2	2	6	2	7	1	2	3	1
3	2	6	7	2	3	3	5	3	2	6	8	4	7	7	2	1	2	1
3	7	4	5	7	2	8	3	3	2	6	6	2	2	3	1	2	1	7
3	2	7	6	3	3	2	6	2	8	2	4	2	5	7	3	7	1	1
3	7	6	2	8	2	3	7	3	6	2	2	3	1	4	2	7	5	1
3	8	7	3	2	3	2	2	6	7	3	5	2	6	1	2	7	1	4
7	5	3	8	6	2	3	3	6	2	3	2	2	7	7	4	1	2	1
3	6	7	2	8	3	6	2	3	2	5	2	7	1	2	7	1	3	4
3	6	3	3	2	6	8	7	2	3	2	5	2	7	4	2	1	1	7
3	7	6	3	2	6	2	7	3	8	7	5	2	2	3	1	2	1	4
5	3	8	7	6	3	3	2	2	7	2	2	6	1	2	1	3	4	7
3	8	3	6	2	2	7	4	3	6	2	2	2	3	1	7	5	1	7
3	3	3	3	2	7	8	6	2	6	2	2	5	7	1	4	2	7	1
3	6	3	7	3	2	4	7	3	6	8	2	7	2	2	2	1	5	1
3	6	2	3	3	3	7	6	7	8	7	2	2	2	2	4	1	1	5
3	8	2	6	2	6	3	7	3	2	7	3	7	5	4	2	1	2	1
3	2	6	3	3	2	2	2	1	7	8	6	2	3	7	7	4	1	5
3	7	2	8	2	7	3	6	3	2	3	5	2	1	6	7	4	2	1
3	2	3	3	3	6	2	2	2	7	5	6	2	7	1	1	8	4	7
3	3	5	6	3	2	2	7	2	8	6	2	3	1	7	2	4	1	7
3	6	7	6	7	8	2	3	2	3	2	5	2	3	2	7	1	1	4
3	6	8	3	3	2	2	2	7	5	2	3	6	4	7	1	2	1	7
6	3	8	3	2	3	7	5	7	7	2	3	4	6	2	2	2	1	1
7	3	7	3	8	7	6	6	4	2	2	3	2	3	2	2	5	1	1

Figura 9: Posibles soluciones.

Aunque el espacio de posibles soluciones en la estructura de los cromosomas sigue siendo bastante amplio, a la hora de graficar el diagrama de Gantt, nos damos cuenta que solo muestra la gráfica de la figura 10.

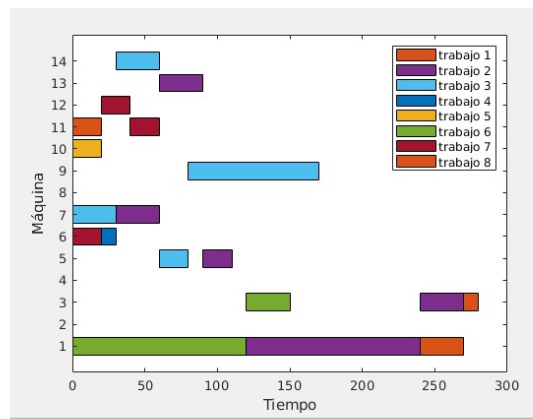


Figura 10: Posible solución Diagrama de Gantt.

Si cambiamos la aptitud, ahora consideramos una aptitud general, donde buscamos disminuir el tiempo en todas las máquinas, el comportamiento cambia, en la figura 11 se observa que la aptitud aumenta, ya que es la inversa de la sumatoria de los diferentes makespan.

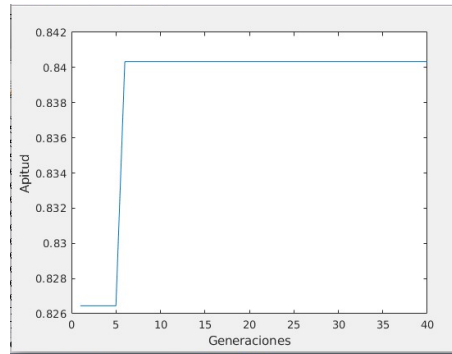


Figura 11: Aptitud por generación.

En la figura 12, se muestra la frecuencia de las generaciones donde encuentra una posible solución, esto tomando en cuenta que se evaluó un makespan general, tratando de encontrar la mejor solución, donde el tiempo general sea el menor.

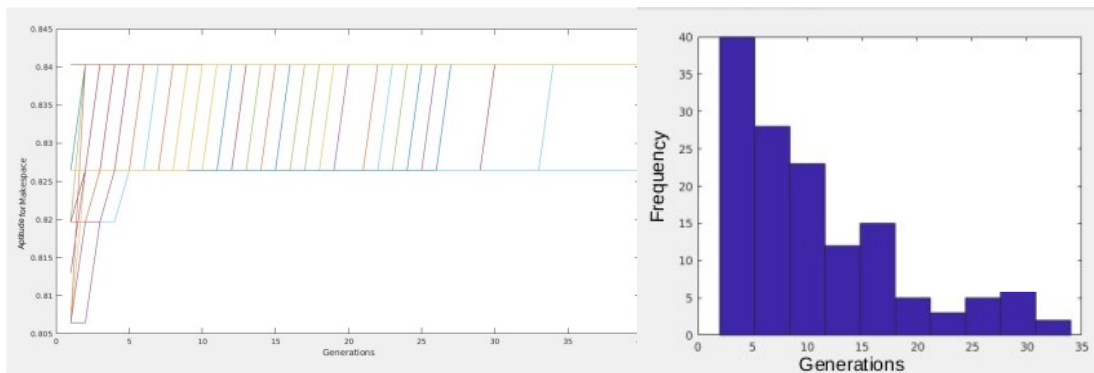


Figura 12: Frecuencia por generación.

7 Discusión de los Resultados.

Ambos análisis, de las dos diferentes maneras de evaluación comienza mostrando el área de posibles soluciones en la estructura de los cromosomas, que aunque estos sean diferentes, su makespan o aptitud es igual, por ejemplo en la imagen 5, observamos alrededor de 40 posibles soluciones, (donde la estructura del cromosoma es diferente), pero si observamos la figura 10, es notorio que las posibles soluciones disminuyen, ya que muchas de estas dan un makespan igual o muy parecido, entonces la variedad de makespan que se pueden generar es menor, pero aún es amplio el área de posibles soluciones, ya que si solo tomamos como evaluación el makespan mayor, en este caso el de 280, podemos observar en la imagen 10, que quien especialmente define este valor de 280, es generado por la tarea 1 operación 2, resaltar que se esto no fuera así, por ejemplo, si terminara con cualquier otra operación, entonces el makespan sería mayor, tomando esto en cuenta, entonces mientras la operación 2 del trabajo 1 se mantenga constante, esto da una pequeña tolerancia para las demás tareas se pueden ejecutar en un cierto rango de tiempo sin que el momento de ejecución de este trabajo se vea afectada.

Entonces, esta tolerancia de tiempo en las demás máquinas que no interactúan o interactúan a un nivel que no afectan la tarea 1, generan un espacio grande de posibles soluciones.

Por otro lado si pasamos a los resultados generados cuando la evaluación de la aptitud, viene dada por un makespan general, observamos en la figura 9, que el área de posibles soluciones en estructura de cromosomas es amplio, podemos suponer por la definición que se da de evaluación que esta área es menor ya que los criterios de evaluación aumentan, por ende menos soluciones son capaces de cubrir o satisfacer estos criterios.

Ahora si observamos la figura 10, el makespan o el tiempo donde se ejecutan los trabajos y sus diferentes operaciones, se mantiene constante, entonces se entiende que los diferentes cromosomas mostrados en la figura 9, aunque son diferentes en arquitectura, el tiempo donde ejecutan las diferentes tareas y sus respectivas operaciones es igual, siendo estos y el diagrama de Gantt generado 10 los que mejor satisfacen los criterios de evaluación general.

Si observamos ambos histogramas de la figura 8 y la figura 12) se muestra que mayormente se encuentra una posible solución en pocas generaciones alrededor de las 5 generaciones, esto inclusive con un número de población pequeño (10 posibles soluciones por generación), pero, es importante recalcar el hecho de que con el primer criterio de evaluación la posible solución se encuentra mayormente antes de la quinta generación, mientras que en el segundo criterio de evaluación, tarda un poco más, esto alrededor de 5 generaciones, esto refuerza el punto expresado en párrafos anteriores, donde se menciona que el área de posibles soluciones es mayor con el primer criterio, y menor con el segundo ya que las características de evaluación aumentan, pero se puede decir que esta diferencia de rigurosidad en la evaluación, genera mejores soluciones.

8 Conclusión.

El gran espacio de posibles soluciones, se debe en parte a la estructura del cromosoma, ya que aunque estos sean diferentes, pueden generar la misma solución, además, cuando las característica de evaluación solo se basa en el mayor tiempo generado en una determinada máquina (en este caso Máquina 1), da la posibilidad de que la ejecución de las demás tareas tenga variaciones, si que el resultado final (en tiempo) se vea afectado.

9 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.

References

- Jebari, K. (2013, 12). Selection methods for genetic algorithms. *International Journal of Emerging Sciences*, 3, 333-344.
- Mattfeld, D. C. (1996). *Evolutionary search and the job shop*. Springer Verlag Berlin.
- 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