

SIMULACIÓN DEL ALGORITMO GENÉTICO

Metodología para la simulación

Con esta simulación del algoritmo genético en una fase anterior a la definitiva donde no se permitirá la asignación de actividades a máquinas paralelas, servirá para ejemplificar cómo funciona el algoritmo y qué información arroja para poder obtener una secuencia factible. También se pretende ajustar de forma experimental los parámetros que se mencionaron en el capítulo anterior.

La siguiente simulación de actividades para un taller con un sistema de producción intermitente se elabora con base en el AG desarrollado en este trabajo, basado en la codificación propuesta por Gen, Tsujimura y Kubota, y con la modificación a esta en su función evaluación propuesta por Gen, Cheng y Tsujimura.

Introducción de la base de datos con tiempos de las operaciones y restricciones

Para extraer la información necesaria para la simulación se parte de las órdenes de trabajo de cada una de las piezas o productos a fabricar de un supuesto lote, estas órdenes son un documento auxiliar que sirve de base para planear la asignación de las actividades con los recursos disponibles en un taller o empresa. Este documento describe las actividades a realizar para los productos, desde el orden de sus operaciones como el tiempo estimado para cada una de ellas. Contiene además otros campos que son indispensables para la retroalimentación que permite estimar mejor los tiempos de operación, y por lo tanto, mejorar la planeación de futuras actividades.

TALLER		NOMBRE PIEZA:		FORMADOR DE PUNZON		
P.C.P		DIBUJO No.	KL458FF	OT	10633	
		CANTIDAD	1	PZA		
		FECHA ALMACEN	23-sep-10	FECHA PISO		
SECUENCIA	DESCRIPCION DE LA OPERACIÓN	TIEMPO ESTIMADO	TIEMPO REAL	NOMBRE DEL OPERADOR	FIRMA DEL OPERADOR	FECHA DE REALIZACION
T.CNC	CAREAR CILINDRAR, VACIAR, DEJAR EXCESO DE MATERIAL PARA RECTIFICAR	4.5				
F.CNC	BARRENAR,	2.5				
BANCOS	REBABEAR, ENVIAR AL ALMACEN	2				
ALMACEN	ENVIAR A TRATAMIENTO TERMICO 53-55 HRC D2	5				
RECTIF	RECTIFICAR DIAMETROS PARA DAR MEDIDAS FINALES	4				
BANCOS	REBABEAR, AJUSTAR PISTA PARA BALERO, EMBALAR	2				

Figura 4.1 Orden de trabajo del trabajo número 3 en la lista de actividades a simular

La información que se tiene en las órdenes de trabajo, se extrae para construir una base de datos en forma de matriz, la cual es necesaria para que el algoritmo en Matlab encuentre una secuencia de actividades factible. La tercera columna indica el número de máquina en la cual se llevará a cabo la tarea. La máquina correspondiente a cada número se indica en la tabla no. 4.1, así por ejemplo, el número 1 indica que la operación se llevará a cabo en un torno CNC, el número 2 corresponde a un centro de maquinado CNC, y así sucesivamente.

Tabla 4.1

T. CNC.	1
F. CNC	2
BANCOS	3
HORNOS	4
RECTIFICADORA	5
T CONV	6
F CONV	7

La cuarta columna en la matriz indica el tiempo en que se estima la duración de la actividad, plasmado en horas. En este tiempo se incluye el de preparación del material a trabajar y del equipo a utilizar, suponiendo que es independiente de la secuencia en que se realicen las actividades.

X=[

1.0000	1.0000	2.0000	7.0000	8.0000	1.0000	1.0000	4.0000
1.0000	2.0000	3.0000	2.0000	8.0000	2.0000	3.0000	2.0000
1.0000	3.0000	5.0000	2.0000	8.0000	3.0000	4.0000	4.0000
1.0000	4.0000	3.0000	1.0000	8.0000	4.0000	5.0000	3.0000
2.0000	1.0000	6.0000	4.50000	9.0000	1.0000	3.0000	2.0000
2.0000	2.0000	3.0000	1.0000	9.0000	2.0000	6.0000	3.0000
2.0000	3.0000	4.0000	5.0000	9.0000	3.0000	4.0000	2.0000
2.0000	4.0000	5.0000	3.0000	9.0000	4.0000	5.0000	3.0000
2.0000	5.0000	3.0000	7.0000	9.0000	5.0000	3.0000	7.0000
3.0000	1.0000	1.0000	4.50000	10.0000	1.0000	1.0000	4.0000
3.0000	2.0000	2.0000	2.5000	10.0000	2.0000	6.0000	2.0000
3.0000	3.0000	3.0000	2.0000	10.0000	3.0000	2.0000	2.0000
3.0000	4.0000	4.0000	5.0000	10.0000	4.0000	5.0000	3.0000
3.0000	5.0000	5.0000	4.0000	10.0000	5.0000	3.0000	1.0000
3.0000	6.0000	3.0000	2.0000	11.0000	1.0000	6.0000	3.0000
4.0000	1.0000	6.0000	2.0000	11.0000	2.0000	3.0000	3.0000
4.0000	2.0000	1.0000	5.0000	11.0000	3.0000	4.0000	6.0000
4.0000	3.0000	2.0000	4.0000	11.0000	4.0000	5.0000	4.0000
4.0000	4.0000	3.0000	5.0000	12.0000	1.0000	1.0000	3.0000
5.0000	1.0000	1.0000	4.50000	12.0000	2.0000	3.0000	1.0000
5.0000	2.0000	5.0000	5.0000	12.0000	3.0000	4.0000	4.0000
5.0000	3.0000	2.0000	2.0000	12.0000	4.0000	5.0000	3.0000
5.0000	4.0000	3.0000	2.0000	12.0000	5.0000	7.0000	2.0000
5.0000	5.0000	4.0000	5.0000	13.0000	1.0000	6.0000	3.50000
5.0000	6.0000	1.0000	2.50000	13.0000	2.0000	7.0000	5.0000
5.0000	7.0000	3.0000	2.0000	13.0000	3.0000	3.0000	3.0000
6.0000	1.0000	3.0000	4.0000	14.0000	1.0000	3.0000	10.0000
6.0000	2.0000	1.0000	3.0000	14.0000	2.0000	6.0000	6.0000
6.0000	3.0000	2.0000	3.0000	14.0000	3.0000	2.0000	6.0000
6.0000	4.0000	3.0000	1.0000	14.0000	4.0000	3.0000	4.0000
7.0000	1.0000	1.0000	9.0000	15.0000	1.0000	1.0000	5.0000
7.0000	2.0000	3.0000	3.0000	15.0000	2.0000	5.0000	5.0000
7.0000	3.0000	4.0000	5.0000	15.0000	3.0000	2.0000	2.0000
7.0000	4.0000	5.0000	9.0000	15.0000	4.0000	3.0000	1.0000
7.0000	5.0000	1.0000	2.0000	15.0000	5.0000	4.0000	3.0000
7.0000	6.0000	3.0000	2.0000	15.0000	6.0000	1.0000	1.5000
				15.0000	7.0000	3.0000	1.0000];

También se requiere para evaluar el tiempo de terminación de las actividades, una matriz para cada uno de los trabajos en la cual se plasmen los tiempos en que se determina el comienzo de cada una de las tareas, se sumen los tiempos de operación conforme se

vayan asignando las actividades y darle un seguimiento a cada uno de los trabajos. En este ejemplo, se muestra la matriz para el trabajo número 1, la cual se compone de 6 columnas, las cuatro primeras con los mismos datos que en la matriz principal, la columna número 5 se declara de principio con ceros, y es en esta columna que el algoritmo irá determinando el tiempo de comienzo de cada actividad dictados por la secuencia obtenida en cada iteración. Por último, en la columna 6 se irán determinando los tiempos de término para cada operación, que no es otra cosa que la suma de la columna 4 y 5. Al final de la corrida del programa estas matrices son las que permitirán establecer en un diagrama de Gantt de manera gráfica la secuencia de actividades, arrojando el tiempo de comienzo y el de término. Para el algoritmo final, que se diseñó para un problema del tipo FJSSP o problema para un sistema de producción intermitente flexible, existirá una 5 columna en la matriz principal que establezca la máquina alternativa en la que se podrá programar la actividad en caso de estar ocupada la máquina principal. También incluirán las matrices de cada trabajo al final una séptima columna que indicará en qué máquina se llevó a cabo el trabajo.

Matriz X_1 que contiene el orden de las actividades para el trabajo número 1.

```
x1 =[
  1.0000    1.0000    2.0000    7.0000         0         0;
  1.0000    2.0000    3.0000    2.0000         0         0;
  1.0000    3.0000    5.0000    2.0000         0         0;
  1.0000    4.0000    3.0000    1.0000         0         0;
  1.0000    5.0000    1.0000         0         0         0;
  1.0000    6.0000    1.0000         0         0         0;
  1.0000    7.0000    1.0000         0         0         0];
```

Matriz X_1 al final de la corrida del AG, esta dicta la programación de cada actividad en una máquina y tiempo determinado.

```
x1 =
  1.0000    1.0000    2.0000    7.0000         0    7.0000
  1.0000    2.0000    3.0000    2.0000   20.0000   22.0000
  1.0000    3.0000    5.0000    2.0000   23.5000   25.5000
  1.0000    4.0000    3.0000    1.0000   41.0000   42.0000
  1.0000    5.0000    1.0000         0         0         0
  1.0000    6.0000    1.0000         0         0         0
  1.0000    7.0000    1.0000         0         0         0
```

SIMULACIÓN DE PROBLEMA JSS DE 15 TRABAJOS EN 7 MÁQUINAS

La herramienta MATHLAB, Optimtool, presenta campos al usuario en los cuales es necesario escoger algún tipo de parámetro de los establecidos previamente, o llamar a alguna función previamente programada. La figura número 4.1 presenta la interfaz del programa.

En el campo número 1 como se indica, se hace referencia a la *Función Evaluación*, que en este caso es la *makespan15*, todas las funciones son llamadas con un símbolo de @.

Inmediatamente en el número 2 se encuentra el campo para introducir el número de variables, que en este ejercicio es la longitud de los vectores soluciones o individuos, que para 15 actividades y 7 máquinas será de 105 elementos.

Del lado derecho de la pantalla, en el número 3, se encuentran los campos de los parámetros que caracterizarán el algoritmo como son Población, Función Escalamiento, Función Selección, Reproducción, Función Mutación, Función cruzamiento y Función Migración. Por último, en el número 4 también se encuentran las opciones para graficar el comportamiento del algoritmo y los criterios de paro.

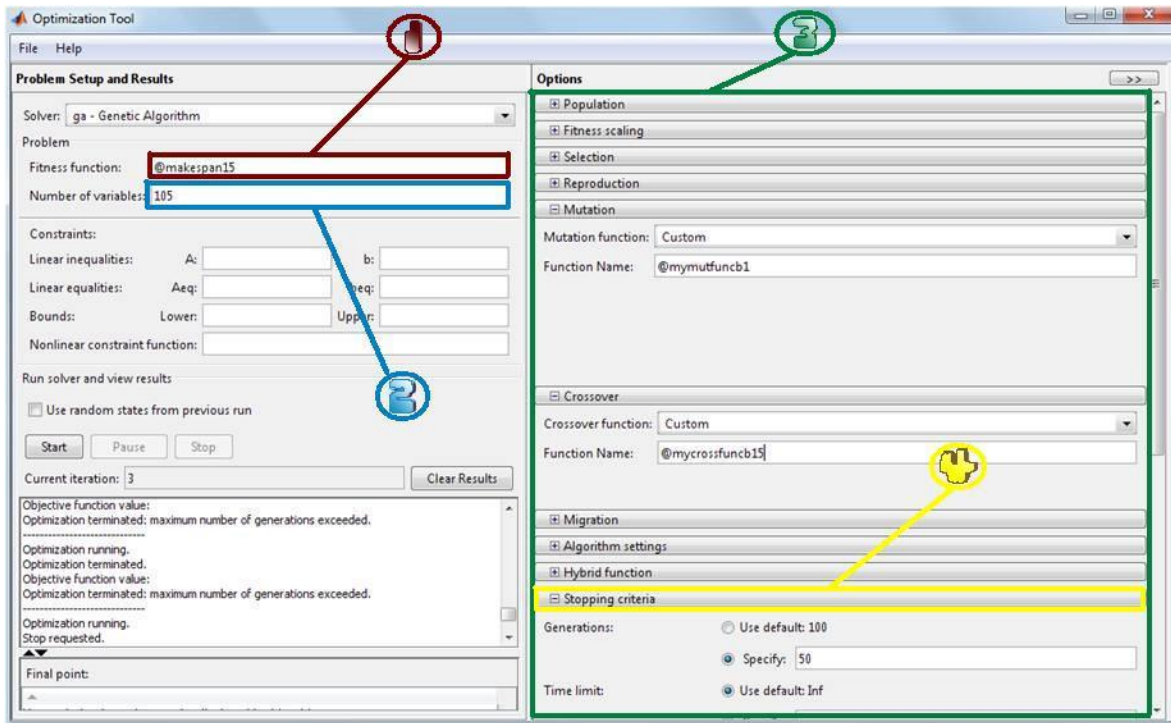


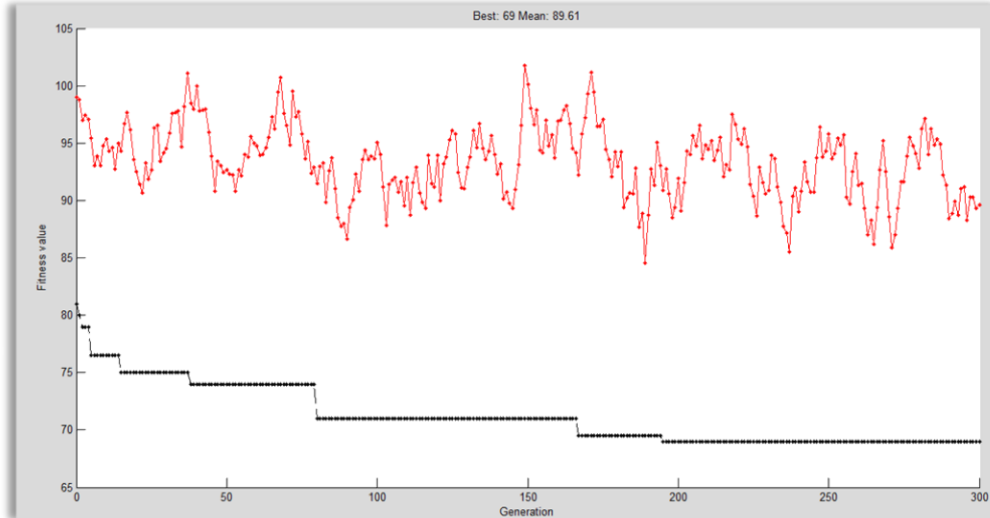
Figura 4.1 Interfaz de Matlab Optintool

Parámetros de la simulación

Tabla 4.2

15 Actividades X7 máquinas				
Criterio de paro	300 generaciones			
f(x) Evaluación-	Makespan15			Población-50
f(x) Escalamiento-	Proporcional			Población inicial - Generada Aleatoriamente
f(x) Selección-	Ruleta			Porcentaje de población para emigrar:20%
f(x) Cruzamiento -	90% de población			Intervalo de Migracion-5
f(x) Mutación -	10% población t>0.8			
Población Elite-	2 individuos			

En la gráfica 4.1 se muestra con línea roja la evolución del parámetro que se busca optimizar representado por la media de la población en cada generación, también se muestra en cada generación el valor del mejor individuo obtenido para cada iteración. El intervalo operativo que arrojo este ensayo fue de 69.



Gráfica 4.1 Desempeño del Algoritmo

Una vez que se ha dado por terminada la simulación se exportan los resultados al *espacio trabajo* de MATLAB, creando el vector solución *optimresults.x* (figura 4.2), ésta es la solución que indica la secuencia con el mejor intervalo operativo. El vector se forma con un valor en cada uno de sus 105 elementos que van desde el número 1 hasta el número 15.

```
>> optimresults.x
ans =
Columns 1 through 13
 14  2  2  2 10 15 11  3  1 13 13 11 10
Columns 14 through 26
 15  5  3  2 10 10  6 14  6 13  2 14  4
Columns 27 through 39
  3 13 11  9  3  1  6  5  9  7 10  7  1
Columns 40 through 52
  9  5 13  8 12 15  4 12  8 15 13 11  2
Columns 53 through 65
 10  6 10  6  4 14  6 13  9  8  3  7  1
Columns 66 through 78
  4  3 14 12  5  9  5  8  8  7 11  3 11
Columns 79 through 91
 12  7  2  7  5  5  6 11 15 15  8 12 15
Columns 92 through 104
  4  4 12  9 12  9  1 14  1  8  7  4  1
Column 105
 14
```

1 Figura 4.2 Vector Solución

Esta secuencia en cada iteración le dicta a la función evaluación la tarea a agendar, y es esta función que se encarga de ir asignando los tiempos de inicio y de término a la matriz que le corresponda a cada actividad, estas matrices que van desde la x1 a la x15 se muestran a continuación:

x1 =	x2 =	x3 =	x4 =
1 1 2 7 0 7 1 2 3 2 32 34 1 3 5 2 34 36 1 4 3 1 47 48 1 5 1 0 0 0 1 6 1 0 0 0 1 7 1 0 0 0	2 1 6 4.5 0 4.5 2 2 3 1.0 10 11.0 2 3 4 5.0 11 16.0 2 4 5 3.0 16 19.0 2 5 3 7.0 21 28.0 2 6 1 0 0 0 2 7 1 0 0 0	3 1 1 4.5 9.0 13.5 3 2 2 2.5 13.5 16.0 3 3 3 2.0 28.0 30.0 3 4 4 5.0 30.0 35.0 3 5 5 4.0 43.0 47.0 3 6 3 2.0 53.0 55.0 3 7 1 0 0 0	4 1 6 2 19 21 4 2 1 5 37 42 4 3 2 4 42 46 4 4 3 5 48 53 4 5 1 0 0 0 4 6 1 0 0 0 4 7 1 0 0 0
x5 =	x6 =	x7 =	x8 =
5 1 1 4.5 13.5 18.0 5 2 5 5.0 22.0 27.0 5 3 2 2.0 28.0 30.0 5 4 3 2.0 55.0 57.0 5 5 4 5.0 57.0 62.0 5 6 1 2.5 62.0 64.5 5 7 3 2.0 66.0 68.0	6 1 3 4 14 18 6 2 1 3 18 21 6 3 2 3 25 28 6 4 3 1 42 43 6 5 1 0 0 0 6 6 1 0 0 0 6 7 1 0 0 0	7 1 1 9 21 30 7 2 3 3 35 38 7 3 4 5 45 50 7 4 5 9 50 59 7 5 1 2 59 61 7 6 3 2 64 66 7 7 1 0 0 0	8 1 1 4 30 34 8 2 3 2 39 41 8 3 4 4 41 45 8 4 5 3 47 50 8 5 1 0 0 0 8 6 1 0 0 0 8 7 1 0 0 0
x9 =	x10 =	x11 =	x12 =
9 1 3 2 30 32 9 2 6 3 32 35 9 3 4 2 35 37 9 4 5 3 40 43 9 5 3 7 57 64 9 6 1 0 0 0 9 7 1 0 0 0	10 1 1 4 0 4 10 2 6 2 11 13 10 3 2 2 16 18 10 4 5 3 19 22 10 5 3 1 34 35 10 6 1 0 0 0 10 7 1 0 0 0	11 1 6 3 4.5 7.5 11 2 3 3 11.0 14.0 11 3 4 6 16.0 22.0 11 4 5 4 36.0 40.0 11 5 2 0 0 0 11 6 2 0 0 0 11 7 2 0 0 0	12 1 1 3 34 37 12 2 3 1 38 39 12 3 4 4 50 54 12 4 5 3 59 62 12 5 7 2 62 64 12 6 3 0 0 0 12 7 3 0 0 0
x13 =	x14 =	x15 =	
13 1 6 3.5 7.5 11 13 2 7 5.0 11.0 16 13 3 3 3.0 18.0 21 13 4 1 0 0 0 13 5 1 0 0 0 13 6 1 0 0 0 13 7 1 0 0 0	14 1 3 10 0 10 14 2 6 6 13 19 14 3 2 6 19 25 14 4 3 4 43 47 14 5 1 0 0 0 14 6 1 0 0 0 14 7 1 0 0 0	15 1 1 5.0 4 9.0 15 2 5 5.0 9 14.0 15 3 2 2.0 30 32.0 15 4 3 1.0 41 42.0 15 5 4 3.0 62 65.0 15 6 1 1.5 65 66.5 15 7 3 1.0 68 69.0	

El la figura 4.2 se muestra la secuencia obtenida en forma gráfica, dictada por los tiempos de inicio y termino de cada una de las matrices anteriores.

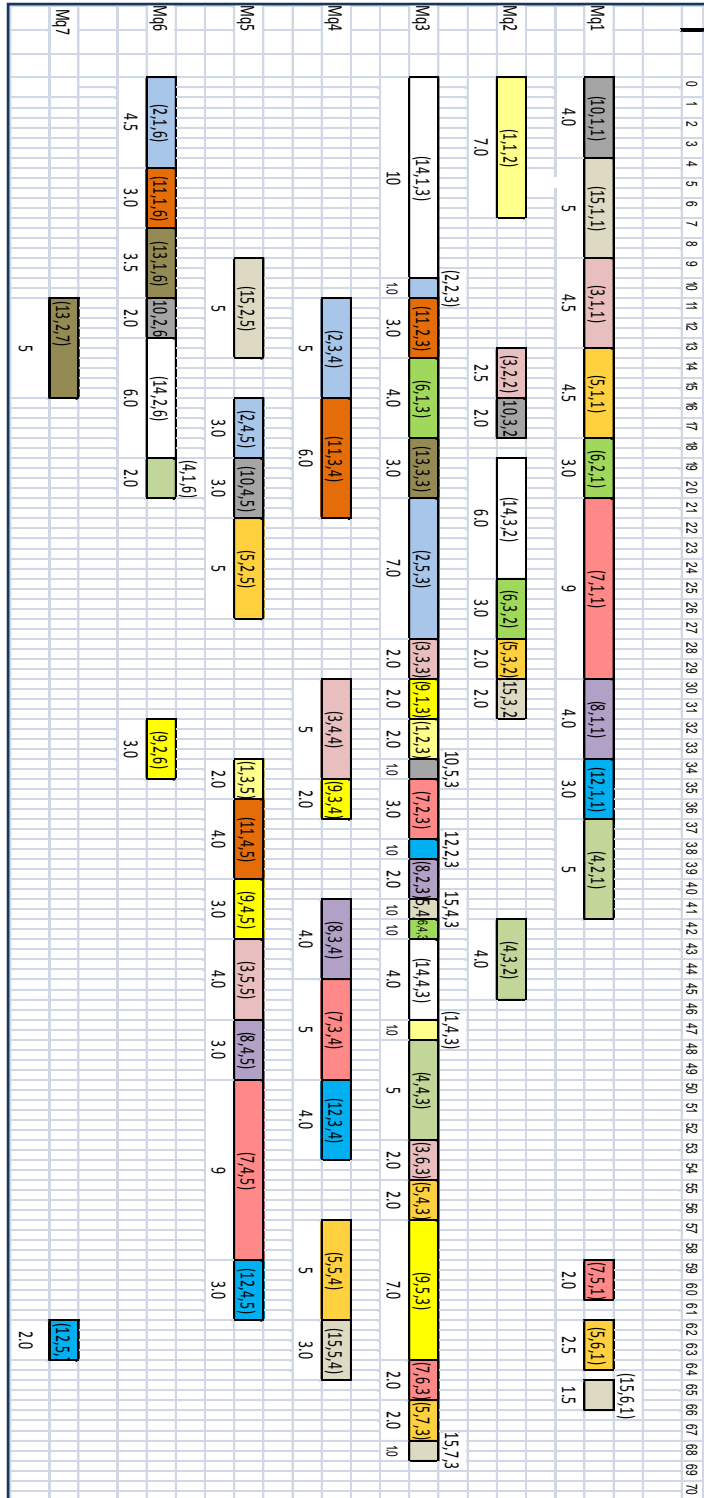


Figura 4.2 Diagrama de Gantt de la secuencia Obtenida

Ajuste de parámetros

Para realizar el ajuste de los parámetros, se realizarán distintos ensayos que permitan visualizar el efecto que éstos tienen en el algoritmo a medida que crecen o disminuyen.

Determinación del parámetro Población

El tamaño de población implica algunos efectos en el algoritmo como son:

Afectan su velocidad; Si el algoritmo realiza en cada iteración la generación de una población en extremo extensa, el algoritmo tendera a ser lento y será menos factible el realizar una búsqueda a lo largo de varias generaciones. Por otro lado, la generación de grandes poblaciones permite el abarcar más espacio en una sola generación, y claro permite la diversidad de las soluciones. Es por esto que se buscará un balance entre una población que permita una búsqueda generalizada, y que no entorpezca la generación de poblaciones nuevas de manera dinámica.

Tabla 4.3 Parámetros AG

20 Actividades X7 máquinas	
Criterio de paro 100 Generaciones	
f(x) Evaluación-Makespan20as	
f(x) Escalamiento-shiftLinear 10	
f(x)Selección-Ruleta	
f(x) Cruzamiento 60% de población	
f(x) Mutación - 40% población $t > 0.1$	
f(x) Migración - 20% Cada 5 generaciones	
Población Elite-variable	

En la tabla 4.3 se establecen las funciones y parámetros utilizados en la simulación con la base de datos con 20 trabajos y 7 máquinas. En la tabla 4.4 se muestran los resultados de 5 ensayos, en los que se asignaron tamaños distintos de población.

Tabla 4.4 Resultados para la simulación de un problema con 20 actividades y 7 máquinas

NÚMERO DE POBLACIONES												
20X7 JSSP	20		40		60		100		140		160	
	Cmáx	Tiempo	Cmáx	Tiempo	Cmáx	Tiempo	Cmáx	Tiempo	Cmáx	Tiempo	Cmáx	Tiempo
1	66	01:09	65.5	01:55	67	02:10	64.5	03:23	65	05:05	65.5	07:20
2	65.5	01:09	67.5	01:55	65.5	02:10	66	03:23	65	05:05	66	07:20
3	67.5	01:09	65	01:55	67	02:10	65	03:23	66	05:05	65.5	07:20
4	66	01:09	67	01:55	65	02:10	65	03:23	66	05:05	64.5	07:20
5	66	01:09	67	01:55	66.5	02:10	64	03:23	63.5	05:05	65.5	07:20
	Elite 2		Elite 4		Elite 6		Elite 10		Elite 14		Elite 16	
PROMEDIO	66.2		66.4		66.2		64.9		65.1		65.4	

Tabla 4.5 Parámetros del AG

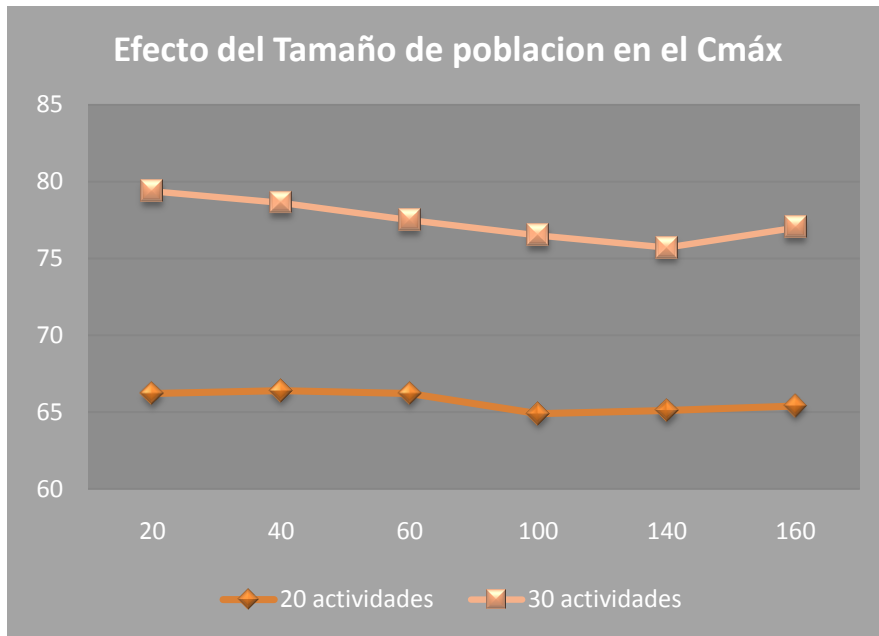
30 Actividades X7 máquinas	
Criterio de paro 50 generaciones	
f(x) Evaluación-Makespan30as	
f(x) Escalamiento-Shift Linear 4	
f(x)Selección-Torneo 20 individuos	
f(x) Cruzamiento 60% de población	
f(x) Mutación - 40% población t>0.1	
f(x) Migración - 20% Cada 5 generaciones	
Población Elite-4 individuos	

En la tabla 4.5 se muestran los parámetros y funciones utilizados en la simulación con la base de datos de 30 trabajos y 7 máquinas. En la tabla 4.6 se muestran los resultados de la simulación para distintos tamaños de población.

Tabla 4.6 Resultado de la simulación para un problema de 30 actividades y 7 máquinas

NÚMERO DE POBLACIONES												
30X7 JSSP	20		40		60		100		140		160	
	Cmáx	Tiempo	Cmáx	Tiempo	Cmáx	Tiempo	Cmáx	Tiempo	Cmáx	Tiempo	Cmáx	Tiempo
1	78.5	00:52	83.5	01:47	79	02:36	78	03:19	77.5	04:51	79.5	05:37
2	79.5	00:52	77	01:47	80	02:36	78	03:19	79.5	04:51	78	05:37
3	79.5	00:52	83	01:47	80	02:36	78	03:19	77.5	04:51	77.5	05:37
4	81.5	00:52	80	01:47	75.5	02:36	75.5	03:19	79.5	04:51	78	05:37
5	81	00:52	78	01:47	78.5	02:36	80	03:19	77	04:51	80	05:37
	Elite 2		Elite 4		Elite 6		Elite 10		Elite 14		Elite 16	
PROMEDIO	80		80.3		78.6		77.9		78.2		78.6	

En la gráfica 4.2 se muestra el comportamiento del Cmáx dependiente del tamaño de población a utilizar para las dos simulaciones de 20X7 y 30X7.



Gráfica 4.2 C_{máx} VS No. Población

En la gráfica 4.3 se muestra el tiempo que se utilizo para cada uno de los ensayos de la tabla 4.4 y 4.6.



Gráfica 4.3 tiempo de simulación VS Población

Se puede apreciar que el desempeño en el parámetro C_{\max} no mejora proporcionalmente con el número de población que se establece para la simulación. En efecto, el espacio solución es muy extenso, por lo que aumentar el tamaño de la población en algunos cientos, no reflejaría ninguna mejoría en el algoritmo. La mejoría del parámetro para la población de 160 contra el resultado que arrojó la simulación con 20 ejemplares, es solo de 1.78% , cuando se incrementó el tiempo en un 548%.

Determinación del criterio de paro; “Generaciones”

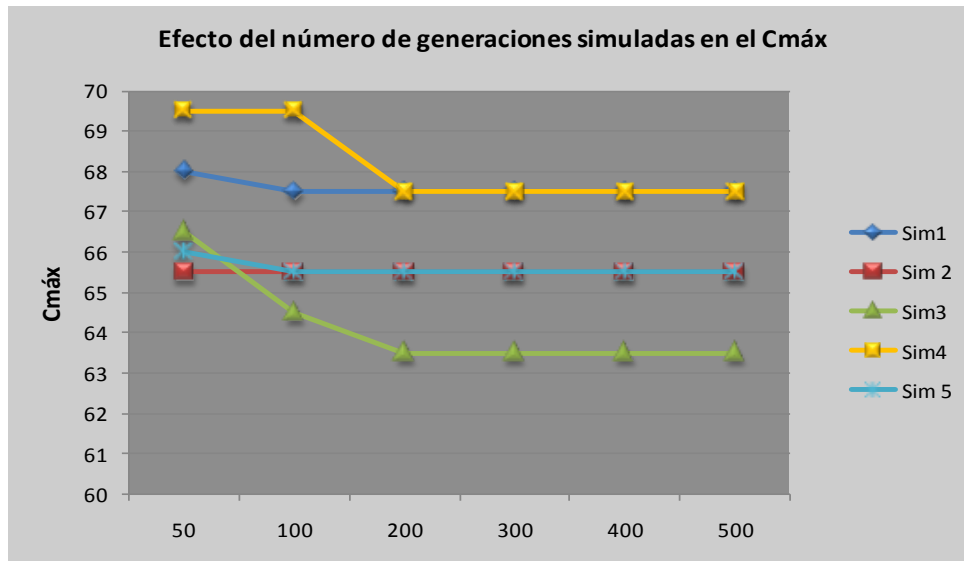
De los criterios de paro que permite elegir la herramienta de *Optimtool*, se escogió el de número de generaciones a simular. Se realizarán los ensayos para dos bases de datos, con el algoritmo programado para observar el desempeño del algoritmo en función del número de generaciones.

Los datos se observan en las tablas número 3 y número 4.

Tabla 4.7 Resultado de la simulación para 20 actividades

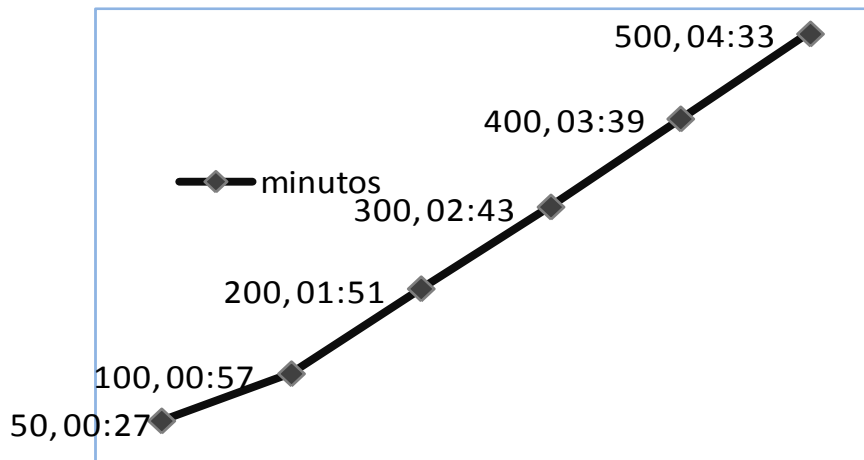
NÚMERO DE GENERACIONES												
20X7 JSSP	50		100		200		300		400		500	
	C_{\max}	Tiempo	C_{\max}	Tiempo	C_{\max}	Tiempo	C_{\max}	Tiempo	C_{\max}	Tiempo	C_{\max}	Tiempo
1	68	00:27	67.5	00:57	67.5	01:51	67.5	02:43	67.5	03:39	67.5	04:33
2	65.5	00:27	65.5	00:57	65.5	01:51	65.5	03:43	65.5	04:39	65.5	04:33
3	66.5	00:27	64.5	00:57	63.5	01:51	63.5	04:43	63.5	05:39	63.5	04:33
4	69.5	00:27	69.5	00:57	67.5	01:51	67.5	05:43	67.5	06:39	67.5	04:33
5	66	00:27	65.5	00:57	65.5	01:51	65.5	06:43	65.5	07:39	65.5	04:33
	Elite 2		Elite 4		Elite 6		Elite 10		Elite 14		Elite 16	
PROMEDIO	67.1		66.5		65.9		65.9		65.9		65.9	

En la gráfica 4.5 se muestran los valores del C_{\max} para cada uno de los ensayos de la tabla 4.7 conforme se avanza en el número de generaciones simuladas.



Gráfica 4.5 C_{máx} VS No. de Generaciones simuladas

La gráfica 4.6 muestra el tiempo transcurrido en la simulación del AG para cada uno de los ensayos de la tabla 4.7. El tiempo para cada ensayo es el mismo, por lo que se muestra solo una gráfica, en cada uno de los puntos se indica en número de generaciones simuladas y el tiempo en el que se alcanzó dicha generación.



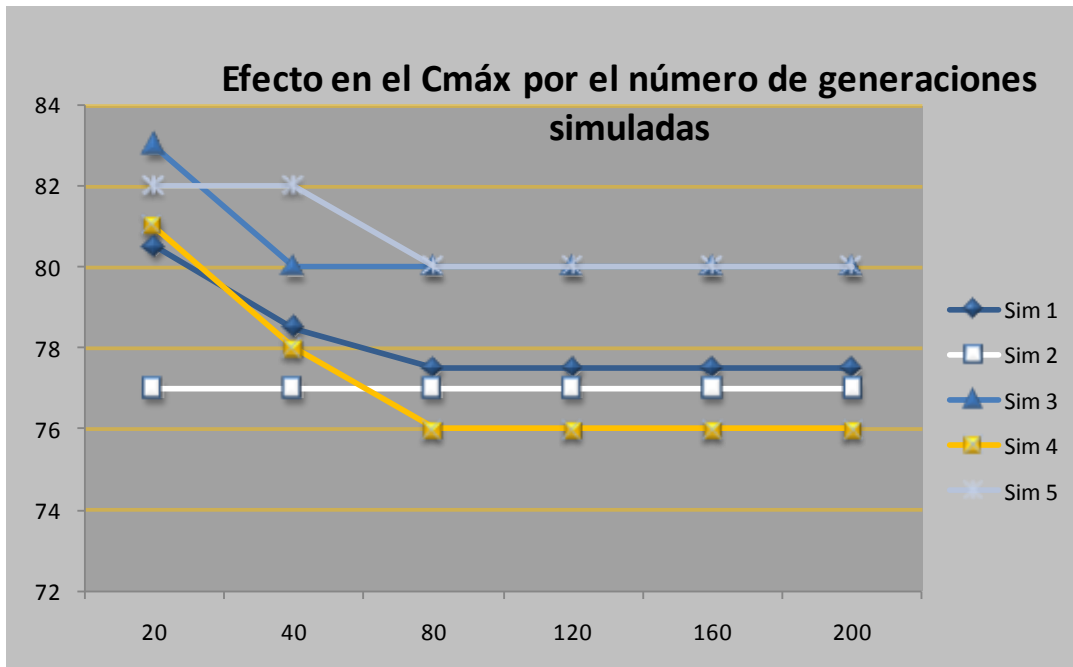
Gráfica 4.6 tiempo VS Generaciones simuladas

En la tabla 4.8 se muestran los resultados de la simulación realizada con el AG para 5 ensayos, los valores de $C_{m\acute{a}x}$ alcanzados en las generaciones 20, 40, 80, 120, 160 y 200.

Tabla 4.8 Resultado de la simulación para 30 actividades.

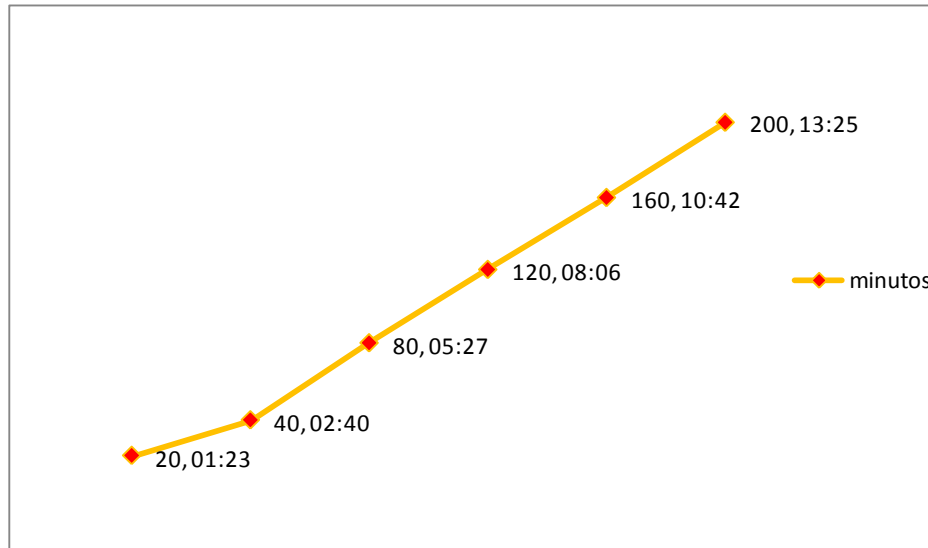
NÚMERO DE GENERACIONES												
30X7 JSSP	20		40		80		120		160		200	
	Cm $\acute{a}x$	Tiempo	Cm $\acute{a}x$	Tiempo	Cm $\acute{a}x$	Tiempo	Cm $\acute{a}x$	Tiempo	Cm $\acute{a}x$	Tiempo	Cm $\acute{a}x$	Tiempo
1	80.5	01:23	78.5	02:40	77.5	05:27	77.5	08:06	77.5	10:42	77.5	13:25
2	77	01:23	77	02:40	77	05:27	77	08:06	77	10:42	77	13:25
3	83	01:23	80	02:40	80	05:27	80	08:06	80	10:42	80	13:25
4	81	01:23	78	02:40	76	05:27	76	08:06	76	10:42	76	13:25
5	82	01:23	82	02:40	80	05:27	80	08:06	80	10:42	80	13:25
	Elite 2		Elite 4		Elite 6		Elite 10		Elite 14		Elite 16	
PROMEDIO	80.7		79.1		78.1		78.1		78.1		78.1	

En la gráfica 4.7 se muestra la evolución de los resultados obtenidos con el AG conforme se incrementa el número de generaciones simuladas.



Gráfica 4.7 Cm $\acute{a}x$ VS no. de Generaciones simuladas

La gráfica 4.8 muestra el tiempo transcurrido en la simulación del AG para cada uno de los ensayos de la tabla 4.8. El tiempo para cada ensayo es el mismo, por lo que se muestra solo una gráfica, en cada uno de los puntos se indica en número de generaciones simuladas y el tiempo en el que se alcanzó dicha generación.



Gráfica 4.8 $C_{m\acute{a}x}$ VS no. de Generaciones simuladas

La gráfica 4.5 indica que después de la población número 200 no se registra ningún cambio aunque el algoritmo continúe con su búsqueda desde el minuto 1:51 hasta en minuto 4:33 como lo indica la gráfica 4.6. Para el segundo grupo de datos, en la gráfica 4.7, esto sucede mucho más temprano. Hasta la generación número 80 se aprecia una mejoría en el parámetro y en adelante, desde el minuto 5:27 hasta el minuto 13:25 según la gráfica 4.8 no se registra ningún cambio. Esta diferencia se debe a que la primera de estas simulaciones se corrió con una población de 20 individuos y la segunda con 60 individuos.

Queda claro que la búsqueda de una buena solución depende de los dos parámetros anteriormente graficados. Las poblaciones pequeñas pueden aportar una solución buena

en muy poco tiempo, que es lo deseable para cualquier problema. El conflicto surge en cuanto la búsqueda con poblaciones pequeñas se estanca en un óptimo local. De estos ensayos que no muestran cambio en ninguna de sus 6 mediciones se encuentra un ejemplo en cada una de las gráficas 4.5 y 4.7.

Para los ensayos finales se simulará considerando una población de 100 como base. No se pretenderá correr el algoritmo por grandes periodos de tiempo puesto que, como muestran los anteriores ensayos, la mejoría del parámetro a la larga representa un esfuerzo computacional muy grande.

Si en alguna de las simulaciones se disminuye este parámetro será debido a que el esfuerzo computacional es muy grande para dicha simulación y no permite el alcanzar un número de generaciones deseado. Se contemplará alcanzar un mínimo de 200 generaciones con poblaciones de 100 individuos.