

# PROGRAMACIÓN DE OPERACIONES EN EL TALLER DE TRABAJO UTILIZANDO LA META - HEURÍSTICA DE RECOCIDO SIMULADO

---

GUSTAVO ALFREDO BULA

*Profesor Auxiliar Escuela de Estudios Industriales y Empresariales  
Universidad Industrial de Santander  
gabula@uis.edu.co*

## RESUMEN

*Se desarrolla una aplicación de la meta-heurística de recocido simulado a la programación en el taller de trabajo para dos configuraciones flow-shop y job-shop. El desarrollo de los algoritmos se hace en el lenguaje de programación Java versión 1.3.*

**PALABRAS CLAVE:** Programación de operaciones, taller de trabajo, flow-shop, job-shop, scheduling, meta-heurísticas, recocido simulado, java.

## INTRODUCCIÓN

La programación de las operaciones en el taller de trabajo hace parte de las decisiones operativas de la planeación y control de la producción. Se han desarrollado diferentes técnicas de modelización de estos sistemas productivos, y diferentes algoritmos matemáticos y computacionales para resolver los modelos. La Programación Lineal Entera Mixta (PLEM), es una de las técnicas de modelización matemática más utilizada, sin embargo dadas las características de problema combinatorio que tiene la programación de operaciones, hace complejo la aplicación del enfoque de PLEM y costoso, en términos de recursos, encontrar la solución óptima del modelo planteado.

Se analiza aquí otra técnica para la programación de operaciones determinística del taller de trabajo, el Recocido Simulado, que permita una mayor flexibilidad en la reprogramación, mayor eficiencia en el uso de recursos y que mantenga o mejore la calidad de la solución hallada por PLEM (en los casos en que está no sea óptima): minimizar la medida de eficacia del Instante de Salida de la Última Pieza del Taller (Makespan).

## PROGRAMACIÓN DE OPERACIONES EN EL TALLER DE TRABAJO

La descripción de los tipos de problemas que se quieren abordar en este análisis: características, términos utilizados y las hipótesis bajo las cuales se trabajan, pueden hallarse en Niño López [1].

## CASOS DE ESTUDIO

Se van estudiar las dos configuraciones productivas que se pueden observar en las Figuras 1 y 2, tomadas del informe sobre el Simulador de DBR propuesto por el Grupo Goldratt.

En la Figura 1 se observa un sistema productivo del tipo "Flow-shop" en donde se fabricarán 8 productos a partir de dos materias primas básicas, las cuales se transformarán a su paso por cuatro centros de trabajo diferentes, siguiendo siempre la misma secuencia.

En la Figura 2 se presenta una configuración del tipo "Job-shop" que elabora 3 productos a partir de 4 materias primas diferentes, en el desarrollo del proceso productivo se encuentran operaciones de ensamble.

Estas dos configuraciones fueron estudiadas por Niño López[2] a través del enfoque de Programación Lineal Entera

Mixta y los valores del makespan encontrados servirán de base para evaluar el desempeño de la aplicación del Recocido Simulado a las mismas configuraciones.

La secuencia de producción se presenta iniciando en la parte inferior hasta la superior, las máquinas están representadas con colores diferentes y los tiempos de preparación de máquina (setup) son independientes de la secuencia

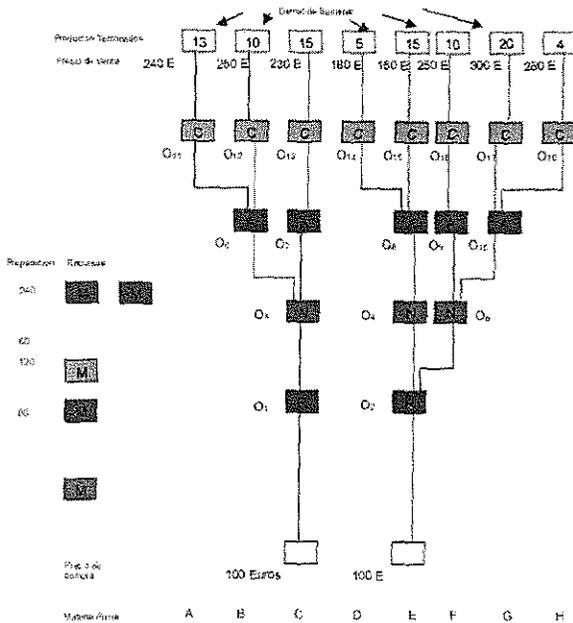


Figura 1. Modelo 1.

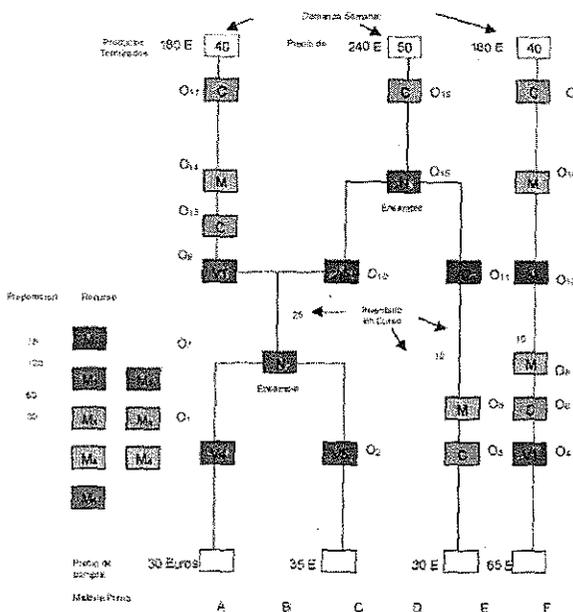


Figura 2. Modelo 2.

### LA METAHEURÍSTICA RECOCIDO SIMULADO

Para Fleisher [3] el recocido simulado es considerado junto a otros esquemas de optimización generalizada como una metaheurística. Su generalidad y aplicabilidad tiene raíces en los fundamentos de la termodinámica y la mecánica estadística. De tal forma que puede ser usado para resolver muchos problemas de optimización combinatoria y algunos problemas con variables continuas.

El algoritmo de Recocido Simulado propuesto por Kirkpatrick et al [4] busca en un conjunto de soluciones, llamado *espacio de configuración*, de la misma forma en que un sistema termodinámico cambia de un estado de energía a otro. El desarrollo del Recocido Simulado establece una analogía entre los problemas de optimización y los sistemas termodinámicos creando así unas nuevas bases para analizar y resolver tales problemas.

Partiendo de una solución inicial, el algoritmo de recocido simulado consta de en una secuencia de iteraciones, en cada iteración la solución se cambia de forma aleatoria partiendo de la solución actual para crear una nueva en el vecindario. El vecindario es definido por el mecanismo de generación seleccionado, este mecanismo es establecido a través de la experimentación.

Fleisher[3] define varios aspectos que deben ser considerados cuando se implementa el algoritmo de Recocido Simulado, antes de que pueda ser dirigido a la solución de un problema en particular. Los más importantes a ser articulados son:

- Un apropiado esquema de enfriamiento
- Una función objetivo adecuada
- Una estructura de vecindario bien definida

Igualmente PHAM y KARABOGA [5] estipulan cuatro decisiones que deben ser hechas para implementar el algoritmo de recocido simulado.

- Representación de las soluciones
- Definición de la función de costos
- Definición de la generación de mecanismos para los vecindarios
- Diseño del programa de enfriamiento

Estos autores definen para el diseño del programa de enfriamiento para el algoritmo de recocido simulado cuatro parámetros a especificar, estos son: temperatura inicial, una regla de actualización de la temperatura, el número de iteraciones en cada temperatura y un criterio de parada para la búsqueda.

El programa de enfriamiento utilizado en muchas implementaciones está basado en una serie finita fundamentada en una temperatura inicial y una temperatura final. En este proyecto se parte de una temperatura inicial,  $T_0$ , que asegure que aplicando el *Criterio de Aceptación de Metrópolis* se acepte un candidato con una probabilidad mayor igual a  $P$ , si este no empeora la función de costo inicial,  $f_0$ , en un porcentaje mayor a  $\alpha$ .

PHAM y KARABOGA [5] explican el criterio de metrópolis como

*"De acuerdo al criterio de Metrópolis, si la diferencia entre el valor de la función de costo actual y la solución nueva generada es igual o mayor que cero, un número aleatorio  $d$  en  $[0,1]$  es generado de una distribución uniforme y si  $\delta \leq e^{-\Delta E}$  entonces, la nueva solución es aceptada. Si no, la solución actual no es cambiada. En la ecuación  $\Delta E$  es la diferencia entre los valores de la función de costo de las dos soluciones". La función de costo está definida aquí como el makespan.*

Se seleccionó la técnica Recocido Simulado, gracias a que la sucesión de soluciones  $\{x_n\}$  generadas por el algoritmo, corresponde a una realización de una cadena Markov, que converge al óptimo global <sup>1</sup>.

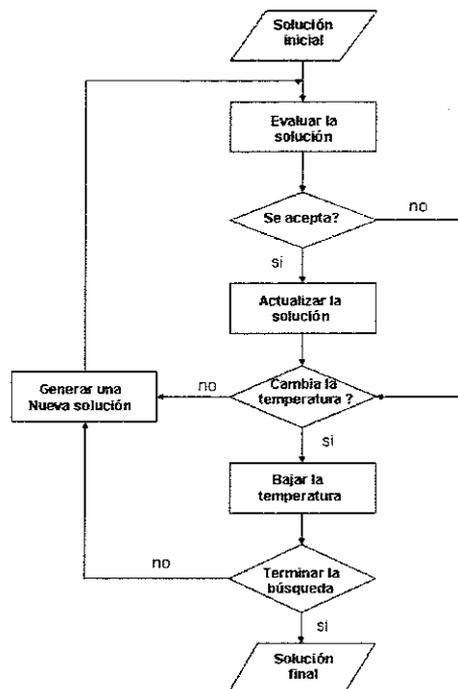


Figura 3. Diagrama de flujo de algoritmo de Recocido Simulado Estándar. Tomado de: PHATN, D.T y KARABOGA, D. Intelligent optimisation techniques. Springer-Verlag. Gran Bretaña. 2000. Pág. 14.

*Primer paso: definir la matriz de solución inicial*

Esto se realiza en el algoritmo *Secuencia*<sup>2</sup>, el cual genera una secuencia aleatoria representada en una matriz  $X$  en donde los vectores filas corresponden a las máquinas (una fila por cada máquina), y los componentes de estos vectores filas corresponden a las operaciones a realizarse en dicha máquina en el orden establecido por el algoritmo.

$$X = \begin{bmatrix} x_{11} & x_{11} & \cdots & x_{1m_1} \\ x_{21} & x_{11} & \cdots & x_{1m_2} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{11} & \cdots & x_{1m_m} \end{bmatrix}$$

Los valores de  $x_{ij}$  representan las operaciones que deben ser realizadas en la máquina  $i$ .

Se parte de una serie de vectores de cadenas de caracteres, uno por cada tipo de máquina, que contienen las operaciones que fueron asignadas a las máquinas, por

### APLICACIÓN DEL ALGORITMO DE RECOCIDO SIMULADO AL PROBLEMA DE PROGRAMACIÓN DE OPERACIONES EN EL TALLER DE TRABAJO

El algoritmo se implementa de acuerdo al diagrama de flujo estipulado en la Figura 3., y se escribe en lenguaje de programación java 1.3.

<sup>1</sup> RÍOS INSUA, David et al. Simulación. Métodos y aplicaciones. Alfaomega, RA-MA. 2000. Bogotá. Pág. 157.

<sup>2</sup> Estos son los nombres de las clases que se escribieron en java

ejemplo para el primer modelo estos son:

```
String[] maq3 = {"01", "02"};
String[] maq4 = {"03", "04", "05"};
String[] maq1 = {"06", "07", "08", "09", "10"};
String[] maq2 = {"11", "12", "13", "14", "15", "16", "17", "18"};
```

Al generar las soluciones (la inicial o nuevas en el vecindario del punto de búsqueda actual) se procede a establecer el orden de las operaciones dentro de cada máquina; habiendo definido previamente que operación va a ser realizada en que máquina.

Para definir en que máquina va a ser realizada una operación se selecciona aleatoriamente una de las máquinas en las que puede ser realizada la operación. Una vez establecido esto se procede aleatoriamente a ordenar las operaciones dentro de cada máquina. Lamentablemente este proceso puede conllevar a que se presente soluciones como la que se muestra en la Figura 4.

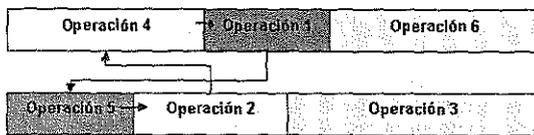


Figura 4. Programación de operaciones con ciclo hamiltoniano presente.

Restricciones de secuencias: Operación 2 precede a la operación 4 en el trabajo amarillo, y la operación 1 precede a la operación 5 en el trabajo azul.

Restricciones de interferencias: La operación 4 debe ser realizada antes que la operación 1, y la operación 5 antes que la 2.

Esta solución produce un ciclo que arrojaría un *makespan* infinito, así que es necesario validar la solución generada de forma aleatoria verificando la existencia de uno de estos ciclos. Para esto la solución se representa a través de un grafo, donde cada nodo representa una operación, y las trayectorias la secuencia que deben llevar las operaciones; ya sea impuesta por el proceso de fabricación o bien por el orden dentro de cada estación de trabajo.

Luego este grafo es representado a través de un matriz  $A_{n \times n}$ , en donde  $n$  es el número de total de operaciones y los elementos de la matriz son unos (1) o ceros (0);  $a_{ij}=1$  si existe una trayectoria de  $i$  a  $j$ , y  $a_{ij}=0$  si no existe (realizado por la clase *Matriz*). Se desarrolló un procedimiento que hace un barrido de la matriz verificando que no se produzcan ciclos hamiltonianos dentro del grafo (realizado por la clase *Validar Ciclos*).

### Segundo paso: evaluar la solución

Evaluar la solución consiste en definir un procedimiento para encontrar el valor de la función objetivo, el *makespan*, para una solución dada. Esto es realizado por la clase *Secuenciación*. Este algoritmo tiene en cuenta el tiempo de inicio más temprano de las operaciones (respetando las restricciones de secuencia e interferencia). El algoritmo recibe la matriz proveniente de *Secuencia* y establece los tiempos de inicio y finalización de cada operación de acuerdo a la secuencia generada aleatoriamente y calcula el valor del *makespan*.

### Tercer paso: generar soluciones en el vecindario del punto de búsqueda actual.

*Definición del vecindario.* En los problemas de programación de las tareas del taller de trabajo hay dos restricciones que limitan el vecindario de soluciones: primero hay una secuencia definida de operaciones para la ejecución de una tarea, y segundo, hay operaciones asignadas a máquinas específicas; así obtenida una solución el vecindario está compuesto por un cambio en la secuencia que las diferentes operaciones asignadas a una máquina tienen. Se probó intercambiar el rango de dos operaciones en la programación en una máquina para proveer una nueva solución.

Este procedimiento es realizado en *Solución*, recibe la matriz de Secuencia e intercambia el orden de dos operaciones en una máquina si esta es seleccionada con probabilidad 0.5 y si existen más de dos operaciones programadas en ella, como se muestra en la Figura 5.

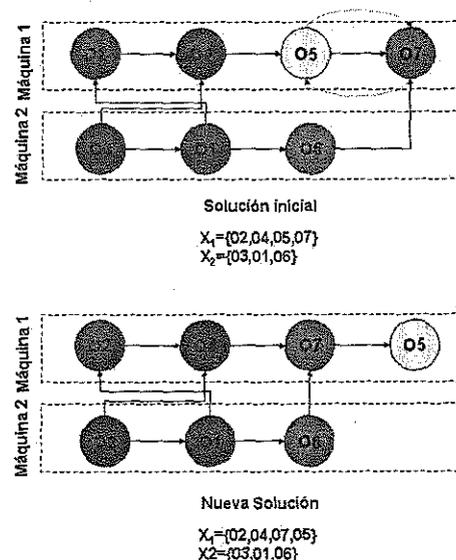


Figura 5. Creación de soluciones en el vecindario de la solución actual.

Cuarto paso: establecer parámetros del recocido simulado.

Tabla 1. Tabla de frecuencias para los resultados generados en el modelo 2.

Los parámetros del recocido simulado fueron establecidos así:

Temperatura de inicio  $T_0$ : La temperatura de inicio se estableció para que haya una probabilidad de 0.9 de aceptar puntos que desmejoren la solución inicial hasta en un 20% del valor inicial.

$$e^{-\left(\frac{\alpha \times f_0}{T_0}\right)} \geq 0.9 \text{ para valores de } \alpha \leq 0.2$$

Esquema de actualización de la temperatura: Se utilizó una regla geométrica, de la forma

$$T_{i+1} = k \times T_i \quad i=0,1,2,\dots$$

El valor de  $k$  se estableció en 0.95.

Condición de equilibrio y parada: Después de varios ensayos se estableció un largo de corrida de 100 iteraciones para cada temperatura (condición de equilibrio); y como condición de parada se determinó que pasados 50 cambios de temperatura, si no se ha encontrado un nuevo óptimo el algoritmo llega a su fin. Con estas características y conociendo la solución óptima a través de PLEM de un modelo de prueba a la cual se llegó, se decidió utilizar los mismos parámetros en los modelos estudiados.

Lo anterior se realiza en **Recocido Simulado**. La solución encontrada se graba en un archivo de texto respuesta*i*.txt, donde  $i$  es el número de la repetición del algoritmo realizada.

El programa principal se llama **Modelo**, este ejecuta cien veces (o el número que se desee) la meta-heurística de recocido simulado y va llevando un registro de los resultados hallados en archivos de texto (recocido*i*.txt), y graba la mejor respuesta hallada en un archivo llamado recocido.txt. De igual forma cada valor del makespan hallado en cada ejecución del algoritmo de **Recocido Simulado** es grabado en el archivo Respuesta.txt para poder medir la eficiencia y eficacia del mismo.

Makespan	Frecuencia	Frecuencia relativa
2317	2	2%
2335	2	2%
2353	1	1%
2359	1	1%
2375	1	1%
2382	1	1%
2390	1	1%
2392	1	1%
2403	2	2%
2404	4	4%
2415	1	1%
2417	1	1%
2443	1	1%
2460	1	1%
2479	2	2%
2480	9	9%
2485	1	1%
2487	1	1%
2492	1	1%
2493	1	1%
2494	2	2%
2497	2	2%
2509	1	1%
2511	1	1%
2512	1	1%
2515	1	1%
2539	1	1%
2629	1	1%
2630	1	1%
2658	1	1%
2794	4	4%
2867	6	6%
2870	6	6%
2883	1	1%
2901	1	1%
3019	6	6%
3020	1	1%
3334	7	7%
3409	4	4%
3410	7	7%
3419	1	1%
3949	9	9%

SOLUCIÓN

Solución del primer modelo.

En la tabla 1. se muestran las frecuencias relativas de las soluciones obtenidas aplicando el algoritmo desarrollado

Para este modelo existen varios óptimos locales, de hecho solo en dos ejecuciones del algoritmo de Recocido Simulado se alcanza el óptimo hallado a través de programación lineal (2317).

La iteración en donde se llega primero a la mejor respuesta es la doce:

El makespan es: 2317.0

**Maquina 1**

0-930 op06 930-1470 op09 1470-2010 op08

**Maquina 2**

110-725 op07 725-2045 op10

**Maquina 3**

595-730 op13 807-937 op12 1374-1474 op16 1474-1599

op11 1890-2050 op17 2050-2122 op18 2122-2212

op14 2212-2317 op15

**Maquina 4**

0-500 op01 500-1160 op02

**Maquina 5**

70-700 op03 700-1440 op05 1440-1800 op04

Las máquinas 1 y 2 corresponden a las dos unidades que se tienen la de la máquina tipo 1, la máquina 3 corresponde a la unidad disponible de maquina tipo 2, la máquina 4 corresponde a unidad disponible de la máquina 3 y la máquina 5 a la de la tipo 4.

Tabla 2. Resultados generados por recocido simulado para el instante de salida de la última pieza del taller del modelo 2.

Máquina	Operación	Inicio	Proceso	Fin
M11	T1.O06	0	930	930
	T6.O09	930	540	1470
	T4.O08	1470	540	2010
M12	T3.O07	110	615	725
	T7.O10	725	1320	2045
M21	T3.O13	595	135	730
	T2.O12	807	130	937
	T6.O16	1374	100	1474
	T1.O11	1474	125	1599
	T7.O17	1890	160	2050
	T8.O18	2050	72	2122
	T4.O14	2122	90	2212
	T5.O15	2212	105	2317
M31	T1.O01	0	500	500
	T4.O02	500	660	1160
M41	T1.O03	70	630	700
	T6.O05	700	740	1440
	T4.O04	1440	360	1800

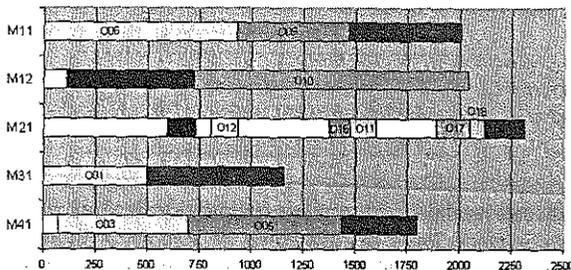


Figura 5. Diagrama de Gantt para el resultado generado por R.S. para el modelo 1.

Solución al segundo modelo.

Debido a que en este modelo se presenta inventario en proceso, las operaciones realizadas sobre estas unidades fueron designadas con números diferentes que las operaciones que se realizan sobre las unidades que hay que procesar desde la primera estación de trabajo.

El máximo valor hallado para el makespan es 3145 y el mínimo valor hallado es 2365, correspondiente a la solución factible encontrada a través de PLEM, con promedio de 2593.43 y desviación estándar de 194.442443.

Tabla 3. Tabla de frecuencias para los resultados generados en el modelo 2.

Makespan	Frecuencia	Frecuencia Relativa	Makespan	Frecuencia	Frecuencia relativa
2365	13	13%	2625	1	1%
2394	2	2%	2640	3	3%
2407	2	2%	2641	1	1%
2419	1	1%	2642	1	1%
2435	2	2%	2644	1	1%
2437	2	2%	2645	1	1%
2439	1	1%	2650	1	1%
2454	9	9%	2651	1	1%
2457	1	1%	2652	1	1%
2462	1	1%	2667	1	1%
2467	2	2%	2682	1	1%
2468	1	1%	2722	1	1%
2469	1	1%	2772	1	1%
2477	1	1%	2782	1	1%
2495	2	2%	2784	1	1%
2505	1	1%	2795	1	1%
2508	1	1%	2801	1	1%
2514	1	1%	2817	1	1%
2527	2	2%	2829	1	1%
2531	1	1%	2830	1	1%
2532	1	1%	2835	1	1%
2534	1	1%	2840	1	1%
2537	1	1%	2854	1	1%
2539	1	1%	2862	1	1%
2555	1	1%	2891	1	1%
2564	1	1%	2915	1	1%
2566	1	1%	2922	1	1%
2572	1	1%	2925	1	1%
2580	1	1%	2932	1	1%
2582	1	1%	2943	1	1%
2595	2	2%	2956	1	1%
2599	1	1%	3019	1	1%
2602	1	1%	3070	1	1%
2617	1	1%	3090	1	1%
2620	1	1%	3145	1	1%

La solución encontrada en la iteración 3 es:

**El makespan es: 2365.0**

*Maquina 1*

0-435 op22      435-590 op23      590-1025 op12  
 1025-1250 op10    1250-1355 op21    1355-2350 op11

*Maquina 2*

0-570 op04      740-1310 op09

*Maquina 3*

0-380 op01      380-650 op20      650-1095 op02

*Maquina 4*

815-1325 op 13    1325-1475 op29    1602-1842 op28  
 1842-2002 op30    2095-2365 op18

*Maquina 5*

162-582 op06      582-792 op24      792-1167 op03  
 1167-1767 op17    1801-2161 op19

*Maquina 6*

204-834 op08      834-1494 op05      1494-1594 op27  
 1594-1824 op25    1824-2064 op16

*Maquina 7*

860-1490 op14

*Maquina 8*

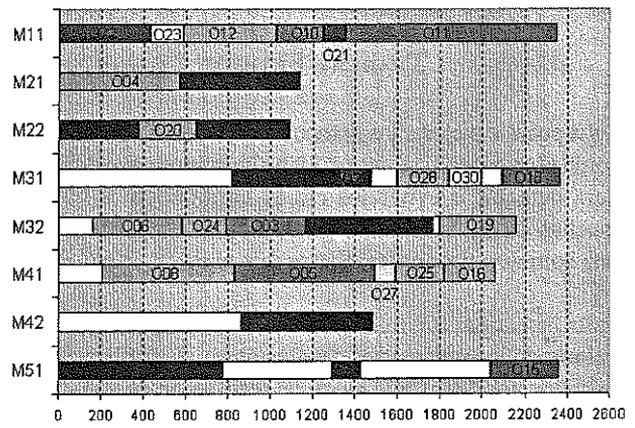
775-1295 op07    1295-1430 op26      2044-2359 op15

La máquina 1 corresponde a la unidad máquina tipo uno; las máquinas 2 y 3 corresponden a las unidades de la máquina dos; las máquinas 4 y 5 a las de la máquina tres; las máquinas 6 y 7 a las máquinas 4 y la máquina 8 a la unidad de la máquina 5.

**Tabla 4.** Resultados generados por recocido simulado para el instante de salida de la última pieza del taller del modelo 2.

Máquina	Operación	Inicio	Proceso	Fin
M11	T2.O22	0	435	435
	T3.O23	435	155	590
	T6.O12	590	435	1025
	T5.O10	1025	225	1250
	T2.O21	1250	105	1355
	T5.O11	1355	995	2350
M21	T6.O04	0	570	570
	T4.O09	740	570	1310
M22	T4.O01	0	380	380
	T1.O20	380	270	650
	T4.O02	650	445	1095
M31	T4.O13	815	510	1325
	T2.O29	1325	150	1475
	T1.O28	1602	240	1842
	T3.O30	1842	160	2002
	T5.O18	2095	270	2365
M32	T6.O06	162	420	582
	T1.O24	582	210	792
	T5.O03	792	375	1167
	T4.O17	1167	600	1767
	T6.O19	1801	360	2161

M41	T6.O08	204	630	834
	T5.O05	834	660	1494
	T3.O27	1494	100	1594
	T1.O25	1594	230	1824
	T6.O16	1824	240	2064
M42	T4.O14	860	630	1490
M51	T4.O07	775	520	1295
	T2.O26	1295	135	1430
	T5.O15	2044	315	2359



**Figura 6.** Diagrama de Gantt para el resultado generado por R.S. para el modelo 2.

**CONCLUSIONES**

A través de la aplicación del recocido simulado a la programación de operaciones del taller se logra que las soluciones converjan a las encontradas a través de PLEM, sin embargo su eficacia no es alta, para a los algoritmos desarrollados para los dos modelos esta fue apenas alcanzó el 2% y 13% respectivamente. Lo que demuestra que hay mayor posibilidad de quedar atrapado en un óptimo local entre mayor sea el número de operaciones analizadas en el problema.

En lo referente a los tiempos de procesamiento para lograr las 100 iteraciones del algoritmo, fue necesario un tiempo de ejecución significativamente mayor para el segundo modelo que para el primero (de tiempos de uno a cinco minutos para el primer se pasa a tiempos de dos hora o más para el segundo), debido al mayor número de operaciones a programar y a que operaciones más avanzadas en el proceso de fabricación utilizan las mismas estaciones de trabajo que las operaciones iniciales; esto posibilita que aumente el número de soluciones en el vecindario de un punto de búsqueda con ciclos hamiltonianos.

En el caso específico del primer modelo, la configuración flow - shop con puntos de divergencia, dada la mayor flexibilidad del Recocido Simulado que la PLEM para adecuarse a las particularidades del modelo, es posible establecer tiempos de inicio más temprano que se adecuen al orden de las operaciones; lo que se traduce en que en los puntos de divergencia la operación, de las operaciones subsiguientes, que sea programada de primero inicie una vez la cantidad necesaria para esto sea producida por la operación precedente. Experimentando al respecto se logró un makespan de 2207 minutos.

- [4] KIRKPATRICK, S., GELATT, C.D. Jr. y VECCHI, M.P. Optimization by simulated annealing, Science, Vol.220, No. 4598, p. 671-680, 1983
- [5] PHAM D.T., KARABOGA D. Intelligent Optimization Techniques. London: Springer - Verlag, 2000

### TRABAJOS FUTUROS

Sería posible mejorar los tiempos de ejecución para modelos como el segundo si se desarrollan algoritmos más eficientes para detectar o evitar los ciclos hamiltonianos, ya que muchas de las soluciones generadas en el vecindario de una solución deben desecharse por ello.

Falta investigar aún más sobre las decisiones acerca de los valores de los parámetros del algoritmo, y la utilización de métodos analíticos que guíen la generación de soluciones en el vecindario.

Aunque se desarrollaron dos conjuntos de algoritmos para cada modelo es posible construir solo un conjunto de ellos para la programación de operaciones del taller de trabajo en general con la interfase adecuada que capture las restricciones de frecuencia e interferencia, tiempos de producción por unidad, número de unidades por trabajo y tiempos de alistamiento por máquina, ya que solo esta información es la necesaria para ejecutar el algoritmo.

### REFERENCIAS

- [1] NIÑO, M. Modelamiento a través de la programación lineal entera mixta del problema de programación de operaciones en el taller de trabajo. Revista de la Facultad de Ingenierías Fisicomecánicas, Vol. 2, No. 1, p. 9-17, 2003
- [2] NIÑO M. Evaluación del Sistema de Programación de Operaciones DBR. [PhD tesis]. Barcelona. Universidad Politécnica de Cataluña. España, 2002
- [3] FLEISHER M. "Simulated annealing: past, present and future". Proceedings of the 1995 Winter Simulation Conference., 1995