ALGORITMO RECOCIDO SIMULADO APLICADO AL PROBLEMA DE SECUENCIAMIENTO DE TAREAS EN SISTEMAS DE PRODUCCION LINEAL FLOW-SHOP

Simulated Annealing Heuristic For Flow Shop Scheduling Problems

RESUMEN

El secuenciamiento de tareas es una labor diaria de muchas empresas del sector de productos y servicios donde se busca optimizar uno o varios objetivos, aquí se propone minimizar el tiempo total de ejecución de todas las tareas. En este documento se presentan los resultados obtenidos a través de la técnica de optimización recocido simulado. Para validar la calidad de las respuestas se usan casos de prueba de la literatura especializada. Se presentan los resultados obtenidos donde se compara la calidad de la solución y los tiempos de ejecución, lográndose resultados satisfactorios y de gran interés académico.

PALABRAS CLAVES: Flow-Shop, heurísticas, Recocido simulado, secuenciamiento de tareas.

ABSTRACT

Task sequencing is a daily job of many companies of the products sector and services, where it is sought to optimize one or several objectives. It is proposed here to minimize the total execution time of all tasks. This document presents the results obtained through optimization technique Simulated Annealing. To validate the quality of the answers, test cases of the specialized literature are used and the results obtained were compared in the quality of the solution and run times. Reaching satisfactory and interesting academic results.

KEYWORDS: Flow-Shop, heuristics, sequencing of tasks, Simulated Annealing.

DAVID ÁLVAREZ MARTÍNEZ
Ingeniero de Sistemas y Computación.
Docente Catedrático
Universidad Tecnológica de Pereira
davidalv@utp.edu.co

ELIANA TORO OCAMPO
Ingeniera Industrial, *M.Sc.*
Docente Asistente
Facultad de Ingeniería Industrial
Universidad Tecnológica de Pereira
elianam@utp.edu.co

RAMÓN GALLEGO RENDÓN
Ingeniero Electricista. *Ph.D.*
Docente Titular
Programa Ingeniería Eléctrica
Universidad Tecnológica de Pereira
ragr@utp.edu.co

1. INTRODUCCIÓN

El problema de secuenciamiento de tareas en sistemas de producción lineal, *Flow-Shop*, se caracteriza porque todos los trabajos a ser programados tienen el mismo flujo de producción aunque varían los tiempos de ejecución en cada una de las estaciones. Para resolverlo se han propuesto diferentes técnicas y metodologías, con el fin de encontrar la secuencia óptima de fabricación cumpliendo con diferentes objetivos como la inexistencia de tiempos muertos de fabricación, reducción de tiempo de cambio y ajustes de las máquinas, anulación de retrasos entre otros, teniendo en cuenta las restricciones propias de cada situación específica como la velocidad de proceso de las máquinas, capacidad de recursos humanos y materiales, etc.

Programar operaciones consiste en el proceso de organizar, elegir y dar tiempos al uso de recursos para llevar a cabo todas las actividades necesarias para producir las salidas deseadas en los tiempos deseados, satisfaciendo a la vez un gran número de restricciones de tiempo y relaciones entre las actividades y los recursos [1].

El problema de *scheduling* implica realizar las siguientes actividades: asignación de trabajos a las máquinas o a los centros de trabajo, determinación de la secuencia de realización de las actividades, programación de las fechas de comienzo y finalización de las operaciones.

Los problemas de secuenciación tienen innumerables aplicaciones en múltiples escenarios tales como: programación de proyectos, operarios, tripulaciones, producción, un grupo de clientes esperando recibir un servicio, un conjunto de programas a ser corridos en un centro de cómputo, entre otros.

En este trabajo se considera el problema de *Flow-Shop,* que se caracteriza por que todos los productivos siguen la misma secuencia de producción; la idea es determinar la mejor secuencia de trabajos y para ello se podría numerar todas las secuencias posibles y elegir aquella que optimiza alguna medida de desempeño.

Por ejemplo para una programación de 23 trabajos implicaría analizar $23! = 2.59*10^{22}$ secuencias posibles, si un equipo pudiera examinar un billón por segundo, tardaría 8.2 siglos para examinarlas todas, eso sin considerar restricciones asociadas a los recursos adicionales (mano de obra y materia prima) y las dependencias entre trabajos (como la preparación), que le dan un mayor grado de dificultad.

De acuerdo a la explosión combinatorial del espacio de soluciones, las técnicas metaheurísticas han mostrado su gran potencial como herramientas de solución para este

Fecha de Recepción: 8 de Septiembre de 2008.
Fecha de Aceptación: 16 de Noviembre de 2008.

tipo de problemas. Aquí se está proponiendo el recocido simulado para la solución del problema *Flow-Shop* considerando como función objetivo la minimización del tiempo total de ejecución de todas las tareas.

La investigación en el campo de secuenciamiento y programación de tareas (*sequencing y scheduling*) ha sido muy amplia. Un panorama excelente en el tema, incluyendo resultados de complejidad computacional, esquemas de optimización exacta y algoritmos de aproximación para diversos tipos de problemas de secuenciamiento puede encontrarse en el trabajo de Lawler et al. [2]. Allahverdi et al. [3] presentan una completísima actualización de problemas de secuenciamiento que involucran tiempos de preparación específicamente; en [4] también se encuentran muchos casos de prueba de variada complejidad matemática con sus resultados y tiempos computacionales, que son ampliamente referidos por grupos de investigadores a nivel mundial, en este trabajo se toma como base este conjunto de casos.

Este documento esta organizado de la siguiente forma: en la sección 2 se describe el problema del *Flow-Shop* y su modelamiento matemático, en la sección 3 se presentan algunas generalidades del recocido simulado, en la sección 4 se presenta el procedimiento empleado para calcular la función objetivo, la estructura de vecindad y la adecuación del algoritmo del recocido simulado al problema especifico, en la sección 5 se presentan los casos de prueba y los resultados obtenidos y finalmente en la sección 6 se plantean las conclusiones, recomendaciones y trabajos futuros con respecto a los problemas de secuenciamiento.

2. DEFINICIÓN DEL PROBLEMA

2.1 Problema de Secuenciamiento de Tareas en Línea

En general, en un problema de *scheduling* intervienen los siguientes elementos: trabajos, disponibilidad, fecha de entrega, tiempo de proceso, prioridad, tiempo de alistamiento (*setup*), operaciones, patrones de flujo y maquinas.

Los objetivos que pueden buscarse pueden ser: minimizar el tiempo de flujo total, minimizar la tardanza total, minimizar el tiempo máximo de terminación (*Makespan*), minimizar la tardanza máxima, minimizar el número de trabajos retrasados, minimizar el retraso máximo.

El problema de *Flow-Shop* se refiere a una configuración enfocada al producto, tal como se muestra en la figura 1.

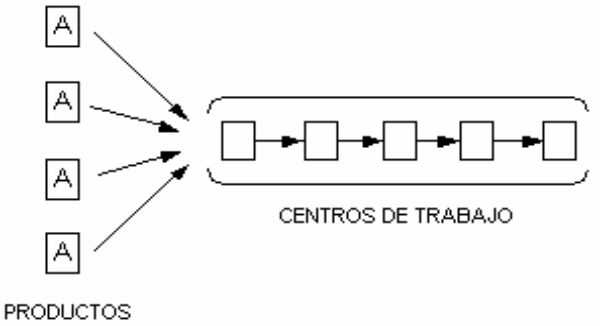

Figura 1. Secuencia de producción lineal.

El problema de línea de flujo considerado consiste en secuenciar n trabajos ($i=1,\ldots, n$) en m máquinas ($j=1,\ldots m$). Un trabajo consiste en m operaciones y la *j-ésima* operación de cada trabajo debe ser procesada en la máquina j, cada tarea tiene el mismo ruteo tecnológico a través de las maquinas. Se asume que cada tarea esta disponible inicialmente y no existe un tiempo límite de entrega. El tiempo de procesamiento de la tarea i en la maquina j se denota por $D_{i,j.}$

Una secuencia de producción bajo está denominación tiene las siguientes características:

- La secuencia de producción de todos los trabajos es igual.
- No se consideran tiempos de ajuste de las máquinas entre un trabajo y otro.
- Cuando se inicia la ejecución de un trabajo en una máquina no puede ser interrumpido.
- Los trabajos tienen que someterse a múltiples operaciones en un número m de máquinas.
- Los tiempos de operación, para de cada trabajo en las m máquinas, pueden ser diferentes.
- El makespan, es un parámetro que indica el tiempo total de ejecución de todas las tareas.

Por tanto el problema del *Flow-Shop* permutacional considera el *makespan* como función objetivo a ser minimizada, resolver el problema significa determinar la permutación que entregue el menor valor de *makespan*.

2.2 Modelo Matemático

Min Z = Makespan(X)

s.a.

$$NTP_j \le 1$$

$$TI_{i,j} \ge TF_{i,j-1} \qquad j = 1,2,\ldots,m$$

$$i = 1,2,\ldots,n$$

$$1 \le x_i \le n$$

$$x_i \ne x_k \forall k = 1,2,\ldots,n-1$$

$$X \in R^n \qquad X \in Z$$

X: Vector de secuencia de producción
x_i*:* Trabajo a realizarse en la posición *i*
$TF_{i,j}$*:* Tiempo de fin de la tarea *i* en la maquina *j*
$TI_{i,j}$*:* Tiempo de inicio de la tarea *i* en la máquina *j*
NTP_j*:* Número de tareas en proceso en la máquina *j*
Makespan: Tiempo de fin de la tarea *n* en la máquina *m*
n y m: Número de tareas y máquinas respectivamente

3. RECOCIDO SIMULADO

El algoritmo del Recocido Simulado (RS) reproduce un conjunto de átomos en equilibrio a una temperatura determinada. En su estructura de vecindad un cambio aleatorio es propuesto para el estado actual y un cambio de energía ΔE es generado, [5] [6].

Si el nuevo estado tiene un nivel más bajo de energía que el estado anterior, $\Delta E \leq 0$, el nuevo estado es tomado para la siguiente iteración. Sin embargo, si el nuevo estado tiene un nivel de energía más alto que el anterior, este es aceptado con una probabilidad $P(\Delta E) = \exp(-\Delta E / k_B T)$, donde T es la temperatura y k_B es la constante de Boltzmann. Esta característica hace que el RS sea diferente a los algoritmos de búsqueda local. El algoritmo RS es presentado en la Figura 2.

El valor de aceptación de un empeoramiento de la energía se mueve inversamente proporcional al ΔE y proporcional a T, la cual decrece con el tiempo. Si se configura inteligentemente el enfriamiento, el algoritmo puede escapar de óptimos locales tempranamente y explorar profundamente otras regiones prometedoras.

RS
1. Escoja una secuencia inicial *S*
2. Halle la temperatura inicial $T = T_0 > 0$
3. Repita:
   a. Escoja una nueva secuencia S'
   b. Sea $\Delta E = E(S)' - E(S)$, donde *E(S)* es la energía (*Makespan*) de la secuencia *S*
   c. Si $\Delta E \leq 0$, acepte la nueva secuencia, haga S=S'
   d. De lo contrario, Si $exp(-\Delta E/T) \geq rand(0,1)$ acepte la nueva secuencia
   e. De lo contrario rechace la secuencia
   f. Reduzca la temperatura de acuerdo al enfriamiento programado
4. Hasta la condición de parada.

Figura 2. Algoritmo del Recocido Simulado

4. PROCEDIMIENTOS

4.1. Codificación del Problema.

4.1.1 Matriz de Tiempos

Los tiempos de duración de cada operación para un trabajo determinado en una máquina se representan mediante una matriz la cual se denomina matriz de tiempos o matriz de duración con la arquitectura mostrada en la figura 3.

La columna *i* de la matriz de tiempos representan los tiempos que tarda en finalizar cada operación en la máquina *j*. La operación del Trabajo 2 en la máquina 3 toma 3 unidades de tiempo, $D_{2,3} = 3$.

	Trabajo 1	Trabajo 2	Trabajo 3	Trabajo 4
Máquina 1	5	1	7	2
Máquina 2	8	7	2	3
Máquina 3	9	3	6	4

Figura 3. Matriz de tiempos.

4.1.2. Secuencia de Trabajo

El orden de entrada de los trabajos a las máquinas se representa mediante un vector, donde la longitud del mismo indica el número de trabajos a procesar y cada elemento dentro del vector representa la tarea que se ejecutará en cada posición, como se observa en la figura 4, leyéndose de izquierda a derecha, el trabajo 2 será el primer trabajo a ejecutarse, el trabajo 4 está en la segunda posición del vector, esto representa que el trabajo 4 entrará en segunda instancia.

Trabajo 2	Trabajo 4	Trabajo 1	Trabajo 3

Figura 4. Vector de ordenamiento

4.1.3. Cálculo del *Makespan*

El calculo del *Makespan* guarda relación con la matriz de tiempos y la secuencia de trabajo, además se debe crear una matriz de tiempos de inicio, el Flow-Shop presenta las siguientes particularidades: El primer trabajo programado solo debe respetar la secuencia tecnológica y la primera máquina nunca tiene tiempo ocioso entre tareas, estas dos propiedades permiten llenar la primera columna y fila de la matriz de inicio, de la siguiente forma:
El tiempo de inicio de la operación del primer trabajo en la máquina *i*, será el tiempo de inicio de la operación del primer trabajo en la máquina *i-1* más el tiempo de duración de la operación del primer trabajo en la máquina *i-1*.
$TI_{i,1} = TI_{i-1,1} + D_{i-1,1}$, donde $1 < i < m$.
El tiempo de inicio de la primera operación del trabajo *j* en la primera máquina, donde $1 < j < n$, será el tiempo de inicio de la primera operación del trabajo *j-1* en la primera máquina más el tiempo de duración de la primera operación del trabajo *j-1* en la primera máquina.
$TI_{1,j} = TI_{1,j-1} + D_{1,j-1}$, donde $1 < j < n$.

Para terminar la construcción de la matriz de inicio se recorre por columnas o filas y el tiempo de inicio $TI_{i,j}$ será el máximo entre:

El tiempo de inicio $TI_{i, j-1}$ más el tiempo de duración $D_{i,j-1}$
El tiempo de inicio $TI_{i-1, j}$ más el tiempo de duración $D_{i-1,j}$

Luego de tener terminada la matriz de inicio el valor del *makespan* es la suma de las dos esquinas inferiores derechas tanto de la matriz de tiempos ordenada como la matriz de inicio, $D_{m, n} + TI_{m, n} = Makespan$.

4.2 Estructura de Vecindad.

La estructura de vecindad para cualquier algoritmo es muy importante, se puede asegurar que es la clave de un buen desarrollo. Luego de analizar el problema de secuenciamiento de tareas y las posibles estructuras de vecindad estudiadas, se logra identificar que los mejores resultados alcanzados se encuentran utilizando troca (*swap*) y corrimiento de tareas.

La troca o *swap* consiste en seleccionar dos tareas cualesquiera de la secuencia (ver Figura 5) e intercambiarlas entre si (ver Figura 6) para obtener un nueva secuencia (ver Figura 7).

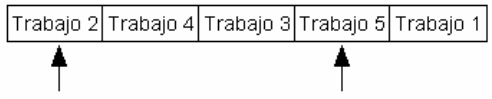

Figura 5. Elección de tareas a trocar.

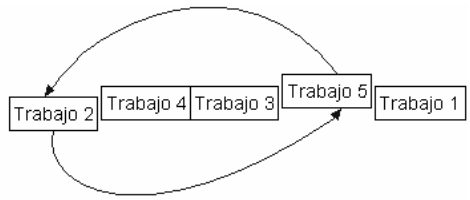

Figura 6. Troca de tareas.

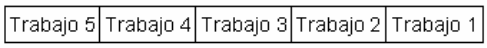

Figura 7. Secuencia después de la troca.

El Corrimiento consiste en seleccionar una tarea al azar de la secuencia (ver Figura 8), escoger una nueva posición aleatoria para la tarea asignada (ver Figura 9) y llevar la tarea escogida a la nueva posición dada (ver Figura 10) obteniendo una nueva secuencia vecina a la anterior (ver Figura 11).

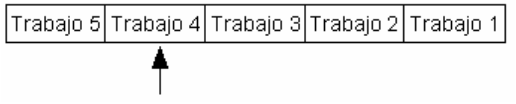

Figura 8. Elección del elemento a ser intercambiado.

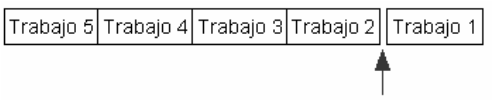

Figura 9. Selección de la nueva posición a ser ocupada.

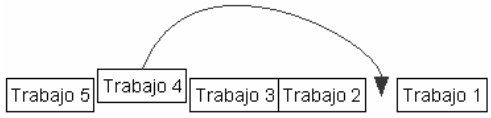

Figura 10. Corrimiento del elemento a la nueva posición.

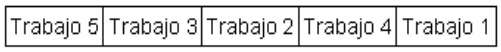

Figura 11. Generación de la configuración vecina.

En este trabajo se definió una estructura de vecindad basada en los procesos de intensificación y diversificación, en el primero se establecen vecinos próximos y en el segundo se efectúan alteraciones significativas para pasar a otras regiones. Se integra una variable aleatoria con distribución normal centrada en 0.5, llamada *Next,* Si *Next* es mayor que 0.5 la elección será la *troca*, de lo contrario el *corrimiento.* El algoritmo utilizado se describe en la tabla 1.

Paso	Acción
1.	Sea *Next = rand(0,1)*
2.	Si *Next >0.5*, Aplique *TROCA*
3.	De lo contrario, Utilice *CORRIMIENTO*

Tabla 1. Algoritmo de elección de la nueva secuencia

4.3 Adecuación del RS para el problema del *Flow-Shop*

El algoritmo del recocido simulado fue escrito en MATLAB ® 7.0 y todo el trabajo computacional lo realizo una máquina con unas especificaciones cercanas a una CPU Pentium(R) 3,0 GHz y 504 MB de RAM.

Como se recomienda en la literatura, se trato de determinar una parametrización única para el algoritmo, que fuera capaz de arrojar resultados satisfactorios para todos los problemas de estudio. La idea central era encontrar una parametrización que fuera lo suficiente fuerte para resolver cada instancia del problema y así, a la vez, probar la robustez del método. Para obtener unos valores eficientes que sirvieran en este articulo se recaracterizaron los parámetros. Los parámetros originales y recaracterizados son dados en la tabla 2. Estas caracterizaciones son un intento de estandarizar los parámetros del algoritmo RS en el problema de secuenciamiento de tareas.

Algoritmo	Parámetros Originales	Parámetros Recaracterizados por
RS	L_o α, β	$L_0 = pm \times N_{trabajos}$ α, β: valores para el enfriamiento programado. $N_{trabajos}$: número de trabajos del problema

Tabla 2. Parámetros originales y parámetros recaracterizados

Para encontrar los valores de los parámetros, se escogieron 3 problemas de diferente tamaño, se crearon parámetros para cada problema y se recombinaron para que existiera una parametrización que presentará las mejores actuaciones. De las recombinaciones realizadas se selecciono una. Los valores de los parámetros resultantes de los experimentos computacionales de la recombinación son presentados en la tabla 3.

Algoritmo	*pm*	α	β
RS	2	0.97	1.06

Tabla 3. Valores finales de parametrización

5. CASOS DE PRUEBA Y RESULTADOS

5.1 Casos de Prueba

Se seleccionaron 30 problemas del *Flow-Shop* de la librería de Éric Taillard del sitio web [4]. El tamaño de los problemas está en un rango de 20 trabajos y 5 máquinas hasta 100 trabajos y 20 máquinas que han sido usados en estudios computacionales anteriores. Los problemas elegidos aleatoriamente se presentan en la tabla 4 y fueron clasificados de acuerdo al tamaño, esta clasificación se presenta en la tabla 5.

Número de Trabajos	Número de Máquinas	Problemas
20	5	ta007, ta008, ta004
20	10	ta012, ta017, ta014, ta020
20	20	ta026, ta022, ta030
50	5	ta032, ta038, ta040
50	10	ta043, ta048, ta049, ta050
50	20	ta052, ta054, ta051
100	5	ta066, ta067, ta070
100	10	ta078, ta077, ta076, ta075
100	20	ta088, ta082, ta087

Tabla 4. Problemas seleccionados aleatoriamente

Problemas Pequeños	Problemas Medianos	Problemas Grandes
ta007	ta032	ta066
ta008	ta038	ta067
ta004	ta040	ta070
ta012	ta043	ta078
ta017	ta048	ta077
ta014	ta049	ta076
ta020	ta050	ta075
ta026	ta052	ta088
ta022	ta054	ta082
ta030	ta051	ta087

Tabla 5. Clasificación de los problemas

5.2 Resultados Computacionales.

En esta sección se presenta la discusión de los resultados obtenidos al ejecutar el algoritmo del recocido simulado para los problemas de Eric's Taillard tomados de la referencia [4], de esta se tomaron 30 problemas en forma aleatoria, los cuales fueron resueltos 20 veces para el algoritmo propuesto. Los parámetros usados son presentados en la tabla 3.

Los resultados obtenidos se presentan en la tabla 6. El análisis se realiza teniendo en cuenta los siguientes conceptos para el cálculo de *PMedio 100%*, *MMedio* y *GMedio* y calculados con la siguiente formula:

$$\sum_{i=1}^{10_{problemas}} \sum_{j=1}^{20_{ejecuciones}} \frac{SolucionObtenida_{i,j} - SolucionOptimaReportada_i}{SolucionOptima_i} * 100\%$$

	Resultados Recocido Simulado
PMedio %	0,550459
MMedio %	3,893617
GMedio %	3,231209
TMedio %	2,558428
Tiempo Ejecución (horas)	52,9336

Tabla 6. Error muestral promedio sobre la solución óptima (porcentaje promedio de 20 ejecuciones para cada problema) y tiempo total de ejecución (en horas).

Siendo *PMedio* el error muestral promedio del conjunto de problemas pequeños, *MMedio* el error muestral promedio del conjunto de problemas medianos y *GMedio* el error muestral promedio del conjunto de problemas grandes. El *TMedio* corresponde al error muestral promedio del conjunto de todos los problemas analizados.

El error muestral es la desviación de la respuesta obtenida con respecto a la mejor solución reportada, los resultados observados en: *PMedio* presentan buena precisión y confiabilidad, *MMedio* muestran una buena precisión y poca confiabilidad, *GMedio* denota que el algoritmo presenta algunas carencias en precisión como en confiabilidad. En el contexto global el algoritmo del recocido simulado presenta un comportamiento satisfactorio.

En los casos de prueba se midió el error muestral promedio con base en la mejor solución dada en [4], el error muestral promedio, la mejor solución Taillard y la mejor solución encontrada por el RS, para los casos de prueba son presentados en la figura 12, la cual valida la calidad y precisión de la técnica del RS implementada para el problema del secuenciamiento de tareas.

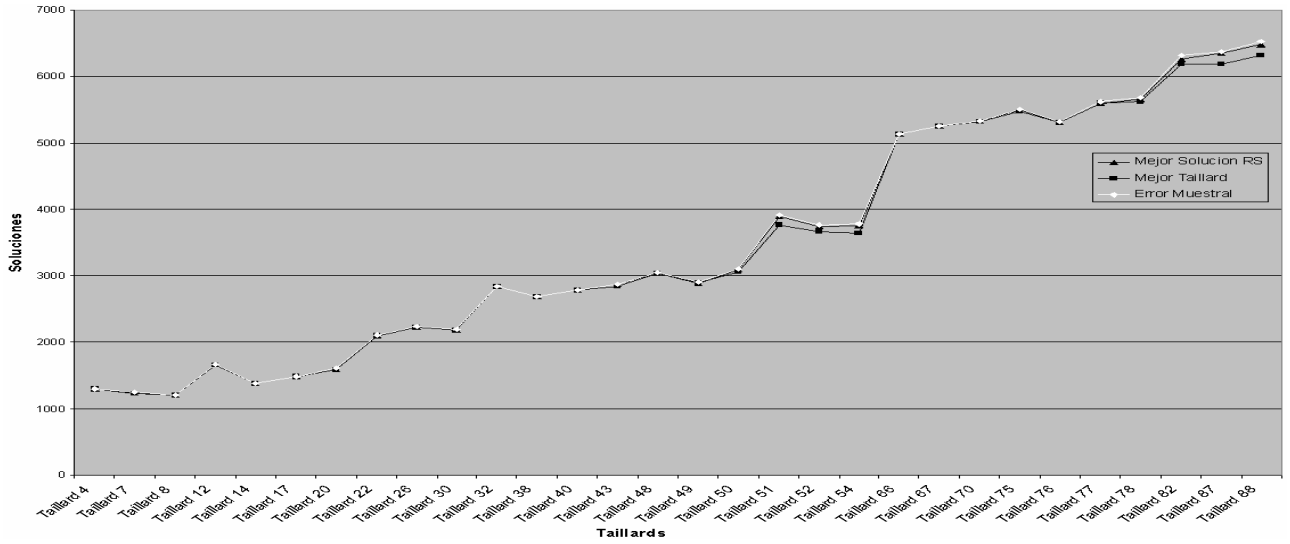

Figura 12. Calidad y Precisión del recocido simulado.

En el proceso de simulación se almacena la mejor configuración con su función objetivo y denominada incumbente. La figura 13 muestra la incumbente durante el proceso de solución de un caso de prueba, el problema 49, tomado de [4]. Cabe anotar que en esta investigación se obtuvieron varios resultados que superan los valores de la librería [4] tal es el caso del problema 49 en el cual se conocía como mejor solución 2897 y en esta se encontró una configuración que reduce el valor 6 unidades dejándolo en 2891.

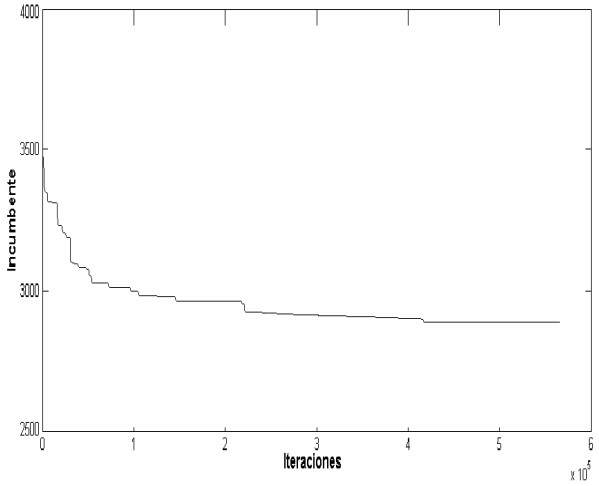

Figura 13. Evolución de la incumbente para un problema en específico.

6. CONCLUSIONES Y RECOMENDACIONES

Se ha resuelto el problema de secuenciamiento de tareas *Flow-Shop* utilizando el algoritmo RS, obteniendo resultados de gran interés académico.

Fecha de Recepción: 8 de Septiembre de 2008.
Fecha de Aceptación: 16 de Noviembre de 2008.

En este estudio se obtuvieron en algunos de los casos de prueba respuestas de mejor calidad a las reportadas en la referencia [4].

Fue implementado el RS con una parametrización única para una gran diversidad de problemas, presentando un comportamiento aceptable, este comportamiento muestra la robustez de la técnica del RS en el problema de secuenciamiento de tareas.

Se propone aplicar optimización multiobjetivo al problema, entre los objetivos sugeridos estarían: tiempo de flujo, tardanza media, *makespan*.

En trabajos futuros se podría aplicar la metodología propuesta al problema de *Job-Shop*.

7. BIBLIOGRAFÍA

[1] Thomas E. Morton, David W. Pentico, "Heuristic Scheduling Systems: With Applications to Production Systems and Project Management". Publicado por Wiley-IEEE, 1993.
[2] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan y D. Shmoys. Sequencing and scheduling: Algorithms and complexity. En S. S. Graves, A. H. G. Rinnooy Kan y P. Zipkin, Vol. 4: Logistics of Production and Inventory, 445-522. North- Holland, New York, 1993.
[3] A. Allahverdi, J.N.D. Gupta y T. Aldowaisan. A review of scheduling research involving setup considerations. *Omega,* 27(2): 219-239, 1999.
[4]http://ina2.eivd.ch/Collaborateurs/etd/problemes.dir/ordonnancement.dir/ordonnancement.html
[5] R. Gallego, A. Escobar, E. Toro, "Técnicas Metaheurísticas de Optimización", Taller de publicaciones Universidad Tecnológica de Pereira, segunda edición: 2008.
[6] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing", Science, vol. 220, pp. 6