

Universidad Politécnica del Estado de Morelos



ANÁLISIS EXPERIMENTAL DE ESTRUCTURAS DE VECINDAD
IMPLEMENTADAS EN UN CALENDARIZADOR PARA UN TALLER DE
MANUFACTURA

T E S I N A

Que para obtener el título de:

INGENIERO EN INFORMÁTICA

P r e s e n t a

WENDY TORRES MANJARREZ

Directores de Tesina

INTERNO: M.C. IRMA YAZMÍN HERNÁNDEZ BÁEZ

EXTERNO: DR. MARCO ANTONIO CRUZ CHÁVEZ

Jiutepec, Morelos

Diciembre de 2010



INGENIERÍA EN INFORMÁTICA

Jiutepec, Morelos a

AUTORIZACIÓN DE IMPRESIÓN DE TESINA

Los abajo firmantes, miembros del jurado para la evaluación del proyecto de estadía de la alumna: **Torres Manjarrez Wendy**, manifiestan que después de haber revisado su tesina titulada “**Análisis experimental de estructuras de vecindad implementadas en un calendarizador para un taller de manufactura**”, realizado bajo la dirección del **M.C. Irma Yazmín Hernández Báez**, el trabajo se ACEPTA para proceder a su impresión.

ATENTAMENTE

MTI. Martha E. Coronel Lemus

Presidenta

Cédula Profesional: 4962692

M.C. Irma Y. Hernández Báez

Secretaria

Cédula Profesional: 4074354

Jiutepec, Morelos

Diciembre de 2010

DEDICATORIA
(pendiente)

TABLA DE CONTENIDO

	PÁGINA
TABLA DE CONTENIDO	I
LISTA DE FIGURAS.....	II
LISTA DE TABLAS.....	III
RESUMEN.....	IV
CAPÍTULO 1.....	1
1.1 Antecedentes	2
1.2 Planteamiento del problema	3
1.3 Panorama general del proyecto.....	4
1.4 Objetivo general	4
1.5 Objetivos específicos	4
1.6 Justificación	4
1.7 Alcances y limitaciones.....	5
1.8 Organización de la tesina	5
CAPÍTULO 2.....	6
2.1 Definición del <i>JSSP</i>	7
2.2 Formulación matemática.....	7
2.3 Modelo del grafo disyuntivo.....	8
CAPÍTULO 3.....	10
3.1 Estructuras de vecindad.	11
3.2 Método de la ruta crítica.....	14
3.3 Estructura de vecindad híbrida.....	15
Bibliografía.	21

ANEXOS

Anexo 1. Título del Anexo
LISTA DE FIGURAS.

Figura	Pág.
Figura 1. Modelo matemático para el JSSP [Roy and Sussman, 64].	8
Figura 2. Modelo del grafo disyuntivo para una instancia de 3x3.	9
Figura 3. Vecindad generada a partir de la solución s.	11
Figura 4. Esquema de movimientos de N1.	12
Figura 5. Esquema de movimientos N4.	12
Figura 6. Esquema de movimientos N5.	13
Figura 7. Esquema de movimientos de N6.	13
Figura 8. Ejemplo del Cálculo de la Ruta Crítica.	14
Figura 9. Algoritmo general de una búsqueda por vecindad.	16
Figura 10. Movimientos realizados por la estructura híbrida seleccionados aleatoriamente.	17
Figura 11. Diagrama general de la estructura de vecindad híbrida.	18
Figura 12. Diagrama de flujo de la estructura de vecindad de la sección correspondiente a la selección del movimiento de vecindad de N1.	19

LISTA DE TABLAS.
(pendiente)

RESUMEN.
(pendiente)

En este trabajo se implementó un algoritmo de búsqueda local con la combinación de cuatro estructuras de vecindad para el problema de calendarización para un taller de manufactura. Los resultados.....

CAPÍTULO 1

Introducción

Este primer capítulo presenta una visión general del proyecto, explicando los antecedentes, planteamiento del problema, objetivos, justificación, así como los alcances y limitaciones del mismo.

1.1 Antecedentes

Las empresas diseñadas para satisfacer pedidos de lotes de producto o servicios de gran tamaño poseen unos procesos productivos altamente repetitivos. En estos casos la calendarización de tareas se suele utilizar para determinar la secuencia productiva de lotes de un mismo producto o servicio que permita cumplir con fechas de entrega de pedidos.

Un taller de manufactura consiste en un conjunto de máquinas diferentes (como tornos, fresadoras, taladros, etc.) que realizan trabajos. Cada trabajo tiene un orden de procesamiento específico a través de las máquinas, es decir, un trabajo se compone de una lista ordenada de las operaciones de cada una de ellas y está determinado por la máquina que requiere y por el tiempo de procesamiento en la misma. La función de la calendarización es la asignación de recursos limitados a tareas a lo largo del tiempo y tiene como finalidad la optimización de una o más funciones objetivo [Noronha and Sarma, 1991].

El problema de calendarización de trabajos en un taller de manufactura o sus siglas en inglés *JSSP*, acrónimo de *Job-Shop Scheduling Problem*, ha sido considerado como uno de los problemas con mayor relevancia en el área de manufactura y la optimización, ya que la administración, así como el manejo eficiente de los recursos, es de vital importancia en las empresas porque permite agilizar procesos, reducir costos y minimizar tiempos [Cruz et al, 2007].

El problema de calendarización de trabajos en un taller de manufactura se define como conjuntos de máquinas y trabajos, donde cada trabajo cuenta con cierto número de operaciones que deben ser procesadas durante determinado tiempo en determinada máquina sin interrupciones. Cada máquina puede procesar únicamente una operación en un instante de tiempo [Cruz et al, 2007].

Este problema ha sido clasificado como *NP-Hard*; una característica del tipo *NP-Hard* es que no se conoce un algoritmo determinístico que resuelva en tiempo polinomial a problemas dentro de esa clasificación [Papadimitriou and Steiglitz, 1998].

Para resolver el problema de calendarización en talleres de manufactura se han utilizado diferentes técnicas para encontrar soluciones cercanas a la óptima en tiempos razonables de cómputo. Una de esas técnicas es la búsqueda local, los métodos de búsqueda local comienzan con una solución del problema y la mejoran progresivamente. El procedimiento realiza en cada paso un movimiento de una solución a otra con mejor valor. El método finaliza cuando para una solución no existe ninguna solución accesible que la mejore.

Una estructura de vecindad nos permite movernos de una solución inicial a una solución vecina mejor. Una estructura de vecindad es un movimiento que se realiza para perturbar la solución inicial de un problema de optimización con el fin de obtener mejores resultados [Papadimitriou and Steiglitz, 1998]. Actualmente se han usado una diversidad de estructuras de vecindad aplicadas a búsquedas locales para resolver este problema. En este proyecto se busca hacer un análisis de algunas (aquellas que han obtenido mejores resultados) estructuras de vecindad para generar una estructura híbrida que combine las mejores características de las estructuras tomadas. El objetivo de la optimización combinatoria es encontrar el máximo (o el mínimo) de una determinada función objetivo sobre un conjunto finito de soluciones.

1.2 Planteamiento del problema

El problema que se pretende resolver es el de encontrar la mejor secuencia de operaciones dentro de un conjunto de máquinas para la realización de un conjunto de trabajos, que minimice el *Makespan*. El *Makespan* es el tiempo de terminación del último trabajo.

Para resolver este problema se propone crear una estructura de vecindad híbrida mediante la combinación de las cuatro mejores estructuras de vecindad que existen en la literatura.

1.3 Panorama general del proyecto

En el área de investigación teórica, existen muchos investigadores que se han interesado por tratar de resolver este tipo de problemas dando como resultado algoritmos muy poderosos que han obtenido muy buenos resultados; en el área real este problema existe en la industria, toda vez que para optimizar recursos es necesario que haya una planificación cuidadosa de los trabajos a realizar [Noronha and Sarma, 1991]. Así que, por lo tanto se puede concluir que este proyecto se encuentra en ambas áreas de interés, ya que si bien es un problema de la vida real, de la industria, también tiene aplicaciones en el área científica, en el campo de la optimización combinatoria.

1.4 Objetivo general

Realizar un estudio experimental de diversas estructuras de vecindad que permita generar una estructura híbrida para aumentar la eficiencia y eficacia en la búsqueda de mejores soluciones al problema de manufactura.

1.5 Objetivos específicos

- Comparar la eficiencia de la estructura de vecindad con respecto a las estructuras de vecindad utilizadas individualmente.
- Comparar la eficacia de la estructura de vecindad con respecto a las estructuras de vecindad utilizadas individualmente.

1.6 Justificación

Con este análisis experimental se busca obtener una estructura híbrida que permita encontrar mejores soluciones para el problema de calendarización de trabajos en un taller de manufactura (*JSSP*). Además de que nos permite agilizar procesos, reducir costos y minimizar tiempos. El estudio de este problema es de suma importancia ya que es un problema práctico del área de la industria de la manufactura, en el cual se busca optimizar los recursos de las empresas dando como resultado el incremento de ganancias y entregas de productos en tiempo y

forma, generando así mas pedidos para este tipo de empresas lo cual repercutiría de manera muy satisfactoria en sus finanzas.

1.7 Alcances y limitaciones

Para hacer este trabajo se utilizarán cuatro estructuras de vecindad representativas de las existentes en la literatura, las cuales han sido catalogadas por obtener los mejores resultados al ser aplicadas a los problemas de optimización. Solo se probará la eficiencia y eficacia del algoritmo híbrido de estructuras de vecindad aplicado al problema de manufactura mediante un análisis de resultados experimentales.

1.8 Organización de la tesina

Esta tesis está dividida en los siguientes capítulos:

Capítulo 1, Comienza con una introducción general del problema de calendarización en talleres de manufactura en el ámbito de la vida real hacia su aplicación en el área teórica.

Capítulo 2, Explicación del problema a través del modelo matemático y del modelo de grafo disyuntivo.

Capítulo 3, Presenta las estructuras de vecindad usadas en este trabajo y el método de la ruta crítica aplicado donde se evalúan estas estructuras, también se presentan los algoritmos de cada estructura de vecindad.

Capítulo 4, Detalla las pruebas experimentales, así como el análisis de los resultados.

Capítulo 5, Presenta las conclusiones de este trabajo.

CAPÍTULO 2

El problema del *JSSP*

En este capítulo se presenta la definición formal y la explicación del problema del *JSSP* a través del modelo matemático y del modelo de grafo disyuntivo.

2.1 Definición del JSSP

El problema del *Job Shop Scheduling (JSSP)*, es considerado uno de los problemas más difíciles en las ciencias computacionales, ya que es clasificado dentro de la categoría de los NP-Complejos [Papadimitriou and Steiglitz, 1998], el estudio de este problema es de suma importancia ya que es un problema práctico del área de la industria de la manufactura. El problema consiste de un conjunto de m máquinas y un conjunto de n trabajos, donde cada máquina lleva a cabo una operación de cada trabajo. Este problema intenta encontrar la secuencia óptima de operaciones en cada máquina que permita la minimización del tiempo máximo de terminación de los trabajos (*makespan*). El modelo que ha sido más frecuentemente usado para aplicarlo al JSSP es el modelo del grafo disyuntivo debido a la facilidad de manipulación [Cruz et al, 2004].

2.2 Formulación matemática

Para el problema de calendarización de trabajos en un taller de manufactura (*Job Shop Scheduling Problem*) con el objetivo de minimizar el *Makespan* se usan los siguientes conjuntos y subconjuntos [Roy and Sussman, 64]:

* Un conjunto de trabajos $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$, un conjunto de máquinas $\mathcal{M} = \{M_1, M_2, \dots, M_m\}$.

* Un conjunto de operaciones $\mathcal{O} = \{1, 2, 3, \dots\}$.

* Subconjuntos de operaciones para cada trabajo ($J_k \subset \mathcal{O}$) y cada máquina ($M_k \subset \mathcal{O}$).

La Figura 1, presenta el modelo matemático para el problema del JSSP.

$\text{Min } (\max_{j \in \mathcal{O}} (s_j + p_j))$	(2.1)
$\forall i, j \in \mathcal{O} \quad s_j \geq 0;$	(2.2)
$\forall i, j \in \mathcal{O}, (i < j) \in J_k$	$s_i + p_i \leq s_j;$ (2.3)
$\forall i, j \in \mathcal{O}, (i, j \in M_k)$	$s_i + p_i \leq s_j \vee s_j + p_j \leq s_i$ (2.4)
$\forall j \in \mathcal{O}$	$s_j + p_j \leq C_{\max}$ (2.5)

Figura 1. Modelo matemático para el JSSP [Roy and Sussman, 64].

La función objetivo es la de minimizar el *makespan*, que definido en base a los tiempos de inicio queda como se especifica en la función (2.1). En esta formulación, los índices i y j hacen referencia a las operaciones. El tiempo de procesamiento de cada operación se representa por p . El conjunto de restricciones (2.2) indican que el tiempo de inicio de cada operación debe ser positivo. El conjunto de restricciones (2.3) que afecta a todo par de operaciones adyacentes i, j de cada subconjunto J_k , define las restricciones de precedencia, el símbolo $<$ significa "precede". El conjunto de restricciones (2.4) define las restricciones de capacidad de recursos, estas restricciones afectan a cada par de operaciones i, j que pertenece a un subconjunto M_k , esto es, si la operación i se ejecuta antes que la operación j , entonces i debe terminar antes de que j inicie su ejecución o bien, si la operación j se ejecuta antes que la operación i , entonces j debe terminar antes de que i comience su ejecución. El conjunto de restricciones (2.5) indica que el tiempo de término de cada operación j debe ser menor o igual que su *makespan* (C_{\max}).

2.3 Modelo del grafo disyuntivo

La figura 2 muestra el modelo del grafo disyuntivo $G=(A, E, O)$ para un JSSP de 3X3 (3 máquinas y 3 trabajos). Este grafo disyuntivo está formado por tres conjuntos; el

conjunto de operaciones O , está conformado por los nodos G , numerados del uno al nueve. El tiempo de procesamiento aparece junto a cada operación. Las operaciones de inicio y fin son ficticias, con tiempo de procesamiento igual a cero. El conjunto A esta compuesto por arcos conjuntivos, cada uno de estos arcos, une un par de operaciones que pertenecen al mismo trabajo. La operaciones 1, 2 y 3 están conectadas por uno de estos arcos y forman el trabajo 1. Los trabajos 2 y 3 están conformados por las operaciones 4, 5, 6 y 7, 8, 9 respectivamente. Cada arco de A representa una restricción de precedencia. El conjunto E está compuesto por arcos disyuntivos, cada arco de E une un par de operaciones que pertenecen a la misma máquina. Se puede apreciar que las operaciones 1, 5 y 7 son ejecutadas por la máquina uno. De la misma manera, las máquinas dos y tres ejecutan las operaciones 2, 6, 8 y 3, 4, 9 respectivamente. Cada máquina forma un cliqué (un subconjunto de E completamente conectado). Cada arco de E representa una restricción de la capacidad de recursos entre un par de operaciones que pertenecen a una misma máquina. Este tipo de restricciones indican que una máquina no puede ejecutar más de una operación en el mismo intervalo de tiempo [Cruz et al, 2004].



Figura 2. Modelo del grafo disyuntivo para una instancia de 3x3.

CAPÍTULO 3

Diseño e implementación de la estructura híbrida de vecindad

Presenta las estructuras de vecindad usadas en este trabajo y el método de la ruta crítica aplicado donde se evalúan estas estructuras, también se presentan los algoritmos de cada estructura de vecindad.

3.1 Estructuras de vecindad.

La vecindad de una solución se define como el conjunto de todas aquellas soluciones que pueden ser alcanzables a partir de una solución s' por medio de un movimiento, un movimiento puede ser un intercambio entre elementos que conforman la solución s [Papadimitriou and Steiglitz, 1998].

$$N(s) = \{s' \in S : s \xrightarrow{\sigma} s'\}$$

Figura 3. Vecindad generada a partir de la solución s .

La Figura 3 presenta la vecindad generada a partir de la solución s , donde $N(s)$ representa la vecindad con respecto a s , s representa una solución tomada del espacio total de soluciones S y s' representa al vecino de s generado a partir de σ (sigma) movimientos [Papadimitriou and Steiglitz, 1998]. Un movimiento puede ser una inserción, eliminación o intercambio de componentes en una solución.

Actualmente se han usado una diversidad de estructuras de vecindad aplicadas a búsquedas locales para atacar este problema. En este proyecto se busca hacer un análisis de diversas estructuras de vecindad para generar una estructura híbrida que combine las mejores características de las estructuras tomadas.

Dichas estructuras de vecindad se describen a continuación.

La Figura 4 presenta el esquema de movimientos de la estructura de vecindad **N1**, introducida por *Van Laarhoven et al* (1992), en esta estructura se propone tomar aleatoriamente un par adyacente de operaciones e intercambiarlas con la restricción de que usen la misma máquina. Esta estructura ha sido probada en algoritmos como el de recocido simulado debido a su fácil implementación y a la calidad de resultados que se obtienen [Blazewicz et al,1996].

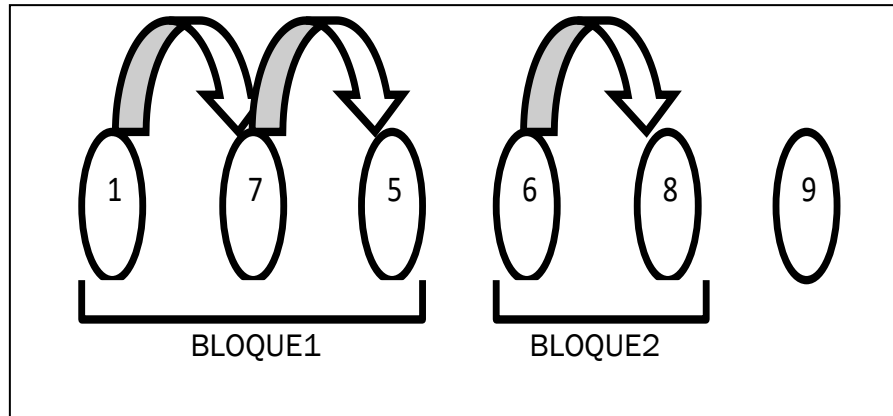


Figura 4. Esquema de movimientos de N1.

En la Figura 5 se muestra el esquema de movimientos de la estructura de vecindad **N4**, propuesta por *Dell'Amico and Trubian*, cuyo movimiento indica tomar una operación de un bloque y moverla ya sea al principio o al final del bloque. Ha sido probada en búsqueda tabú obteniéndose muy buenos resultados [Dell'Amico and Trubian, 1993].

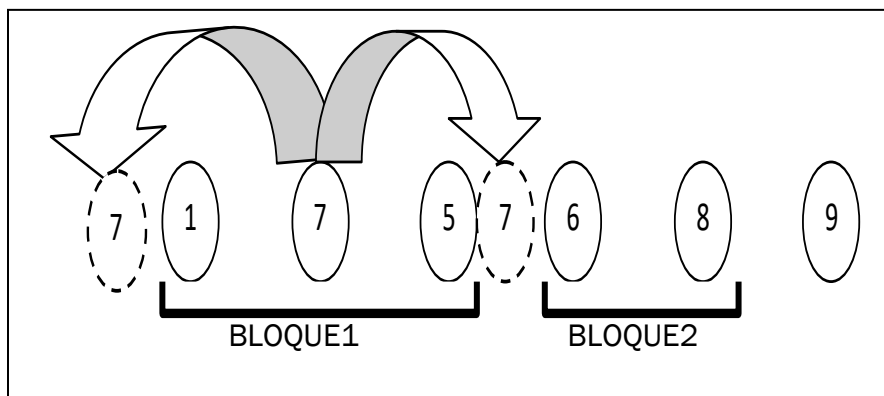


Figura 5. Esquema de movimientos N4.

La figura 6 muestra los movimientos a efectuar por la estructura de vecindad **N5**, creada por *Nowicki and Smutnicki*, propone el uso de bloques como agrupamientos para operaciones que se efectúen en la misma máquina, ahí los movimientos son

como sigue, en el primer bloque de operaciones, intercambia el último par de operaciones, en el bloque siguiente, intercambia el primer par de operaciones del bloque. Aplicada en algoritmos de búsqueda Tabú, es la que mejores resultados ha obtenido [Blazewicz et al,1996].

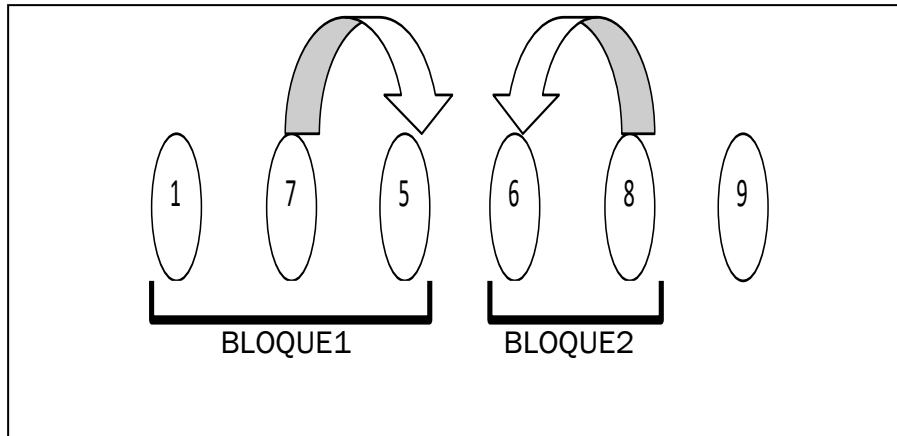


Figura 6. Esquema de movimientos N5.

En la Figura 7 se muestran los movimientos que realiza la estructura de vecindad **N6**, propuesta por *Balas and Vazacopoulos*, se toma una operación que no sea la primera ni la última del camino crítico y se intercambia hacia adelante, después se toma otra operación y se intercambia hacia atrás del camino crítico. Esta estructura ha sido aplicada en el algoritmo de Búsqueda local guiada, obteniendo resultados óptimos para instancias grandes [Blazewicz et al,1996].

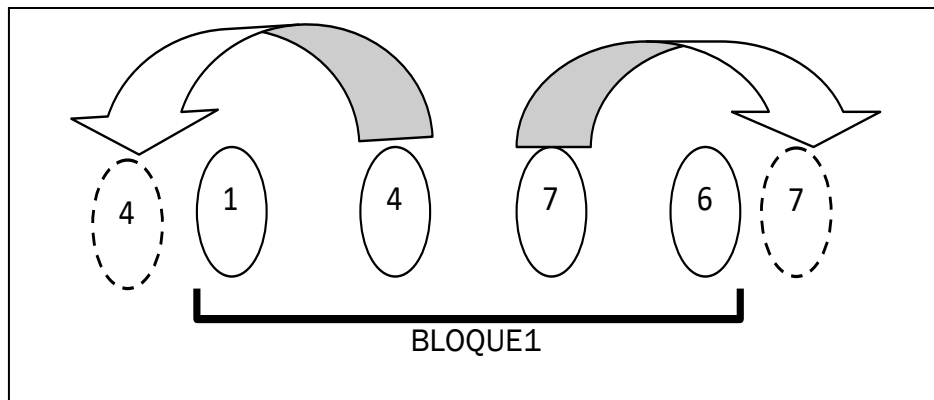


Figura 7. Esquema de movimientos de N6.

Estas cuatro estructuras se combinarán formando una estructura híbrida que se probará en un algoritmo de búsqueda local y se analizarán los resultados obtenidos, evaluando la eficiencia y eficacia del algoritmo.

3.2 Método de la ruta crítica

El método de la ruta crítica se utiliza en el campo de la investigación de operaciones para la planeación y el control de proyectos. En este caso lo usaremos para identificar la ruta crítica y las operaciones que la conforman, de tal manera que, acto seguido, se pueda identificar los bloques críticos de operaciones, cabe mencionar que un bloque crítico es un conjunto de operaciones que se ejecutan en la misma máquina.

El algoritmo de CPM (*Critical Path Method*), inicialmente obtiene datos de un dígrafo, cada arco del dígrafo representa la actividad llevada a cabo por la operación que la precede. Cada una de las actividades es ejecutada en un tiempo t , que es igual a la duración del proceso de la operación que le precede, [Cruz et al, 2004].

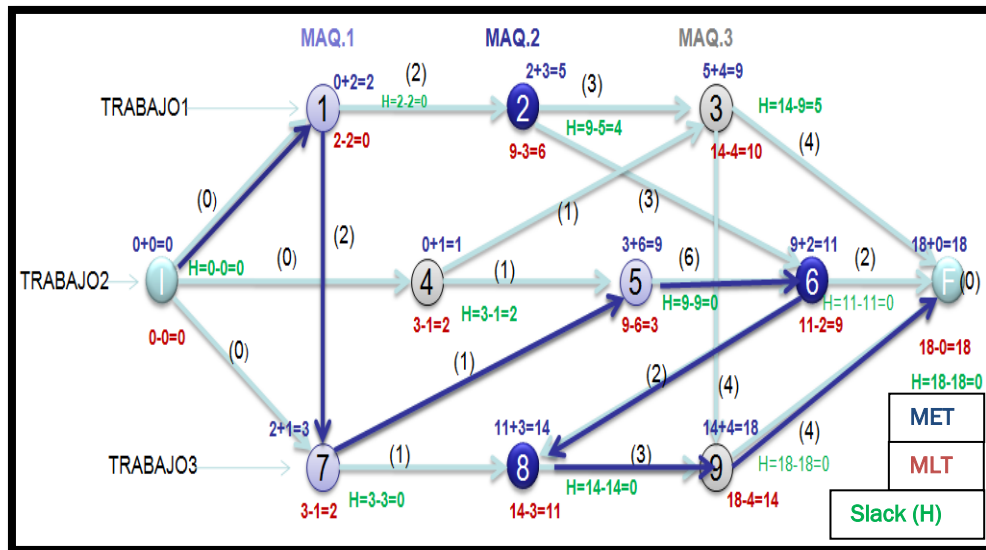


Figura 8. Ejemplo del Cálculo de la Ruta Crítica.

La Figura 8 muestra un ejemplo del Cálculo de la ruta crítica, el algoritmo CPM pasa a través de cada nodo del grafo dos veces, una hacia adelante, de la operación *I* hacia la operación *F*, y una hacia atrás, de la operación *F* hacia la operación *I*. En la pasada hacia adelante a través del grafo, a cada operación se le asigna un Máximo Tiempo más próximo (*Maximum Earliest Time*), ($MET = ET + t_{act}$), donde el tiempo más próximo *ET* (*Earliest Time*) de la operación es el tiempo en el cual la operación empezará si las actividades precedentes son llevadas a cabo lo más rápido posible. En la pasada hacia atrás a través del grafo, a cada operación se le asigna un Mínimo Tiempo más lejano (*Minimum Latest Time*), ($MLT = LT - t_{act}$), donde el tiempo más lejano *LT* (*Latest Time*) de la operación es igual al último momento en el cual la operación podría empezar sin demora en la finalización de cualquier otra operación. Después de calcular el Máximo Tiempo más próximo y el Mínimo Tiempo más lejano, se calcula la diferencia y al resultado se le llama Holgura (*Slack*). La ruta crítica está conformada por todas aquellas operaciones las cuales en su holgura tienen un valor de cero [Cruz et al, 2004].

3.3 Estructura de vecindad híbrida.

Un aspecto muy importante en el diseño de algunos algoritmos de optimización es la elección de una estructura de vecindad adecuada, es decir, elegir aquella estructura de vecindad que permita la mejor explotación del espacio de soluciones de acuerdo al algoritmo utilizado y al problema tratado.

Para determinar el funcionamiento de una estructura de vecindad, se debe definir el criterio de selección de un vecino, es decir, el tipo de movimiento que se llevará a cabo para alcanzar una solución *s'* desde una solución *s*, de modo que si el criterio de selección se cumple se lleva a cabo el movimiento, el proceso se repite hasta que la solución encontrada no pueda ser mejorada.

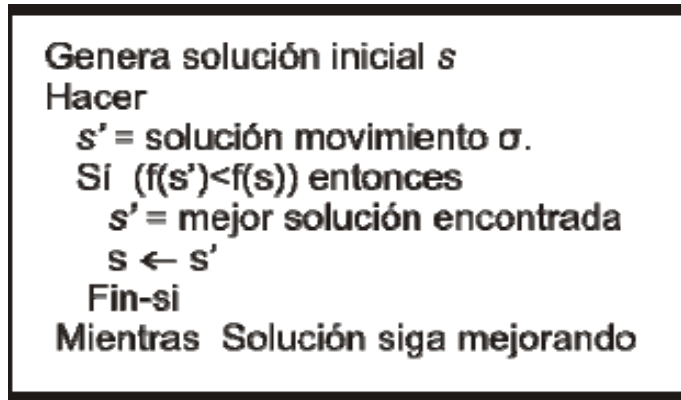
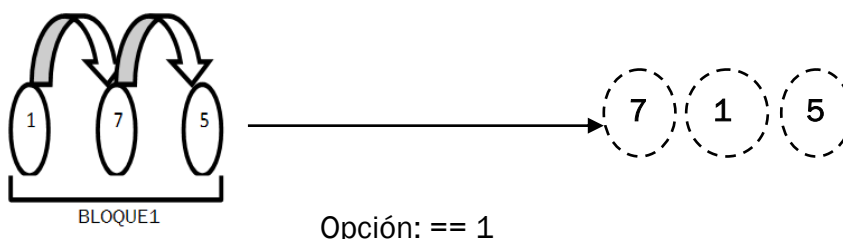


Figura 9. Algoritmo general de una búsqueda por vecindad.

La Figura 9, muestra el algoritmo general de una búsqueda por vecindad, en donde, s es una solución inicial, s' una nueva mejor solución, σ es el movimiento a aplicar para mejorar la solución, es un proceso iterativo que se efectúa hasta que no se encuentra una mejor solución.

Para desarrollar una estructura híbrida de vecindad, se buscó en la literatura aquellas estructuras de vecindad que contaran con buenos resultados en su aplicación. La estructura de vecindad híbrida propuesta en este trabajo es una mezcla de las estructuras de vecindad explicadas anteriormente, en ella se incorporan las ventajas de cada una de las cuatro estructuras de vecindad seleccionadas; N1, N4, N5 y N6.

El tipo de movimiento se escoge aleatoriamente durante la ejecución del algoritmo. Al inicio del procedimiento realizado por la estructura híbrida de vecindad se obtiene un vector con las operaciones que forman la ruta crítica dividida en bloques, un bloque es el conjunto de operaciones que se efectúan en la misma máquina, después de este paso se procede a elegir de forma aleatoria el tipo de estructura a aplicar.



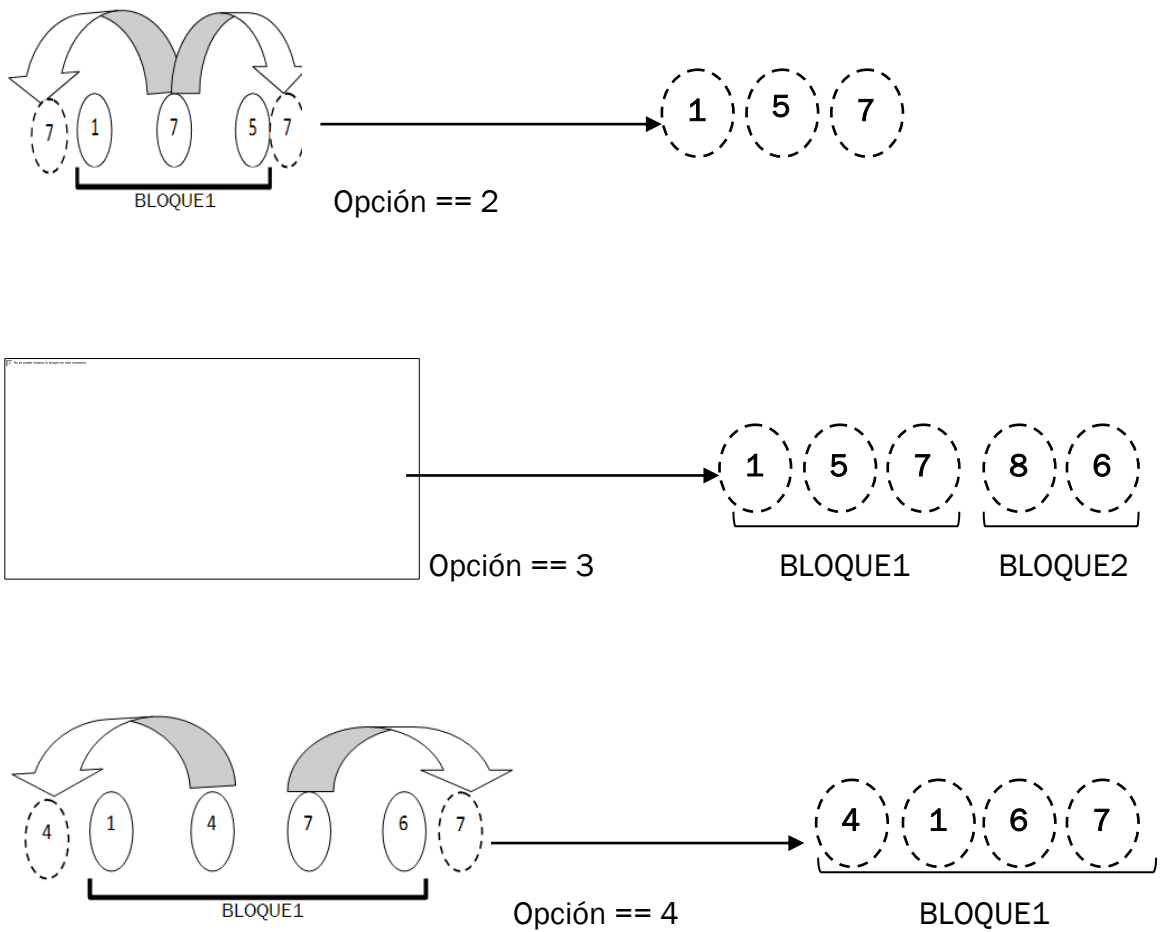


Figura 10. Movimientos realizados por la estructura híbrida seleccionados aleatoriamente.

La Figura 10 muestra las permutaciones realizadas en la estructura de vecindad híbrida, cada permutación es elegida aleatoriamente. Si el valor de la variable opción == 1, el tipo de permutación que se efectúa es el intercambio adyacente correspondiente a la estructura de vecindad N1, opción == 2, intercambio correspondiente a N4, opción == 3, permutación correspondiente a N5, opción == 4, permutación correspondiente a N6.

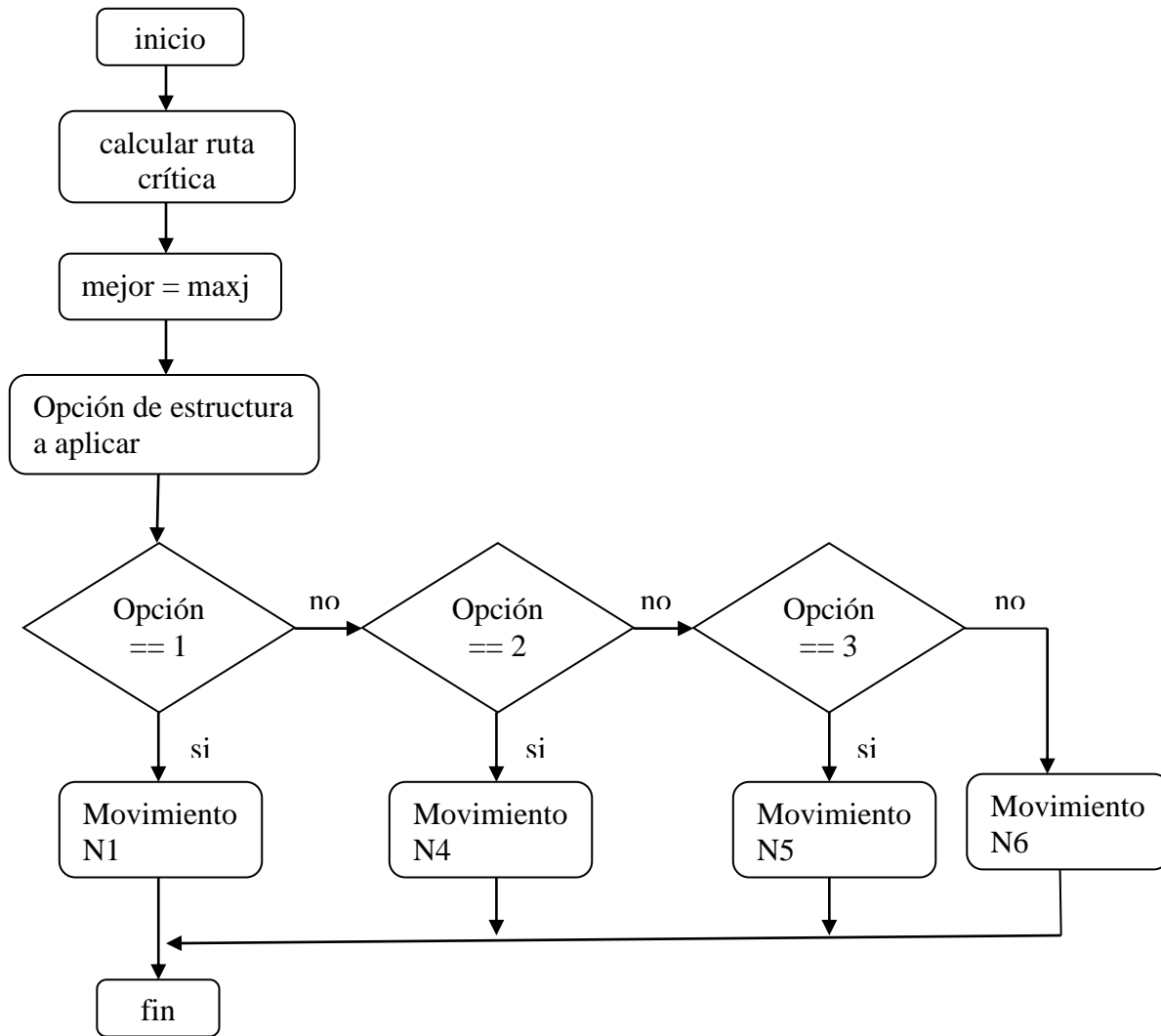


Figura 11. Diagrama general de la estructura de vecindad híbrida.

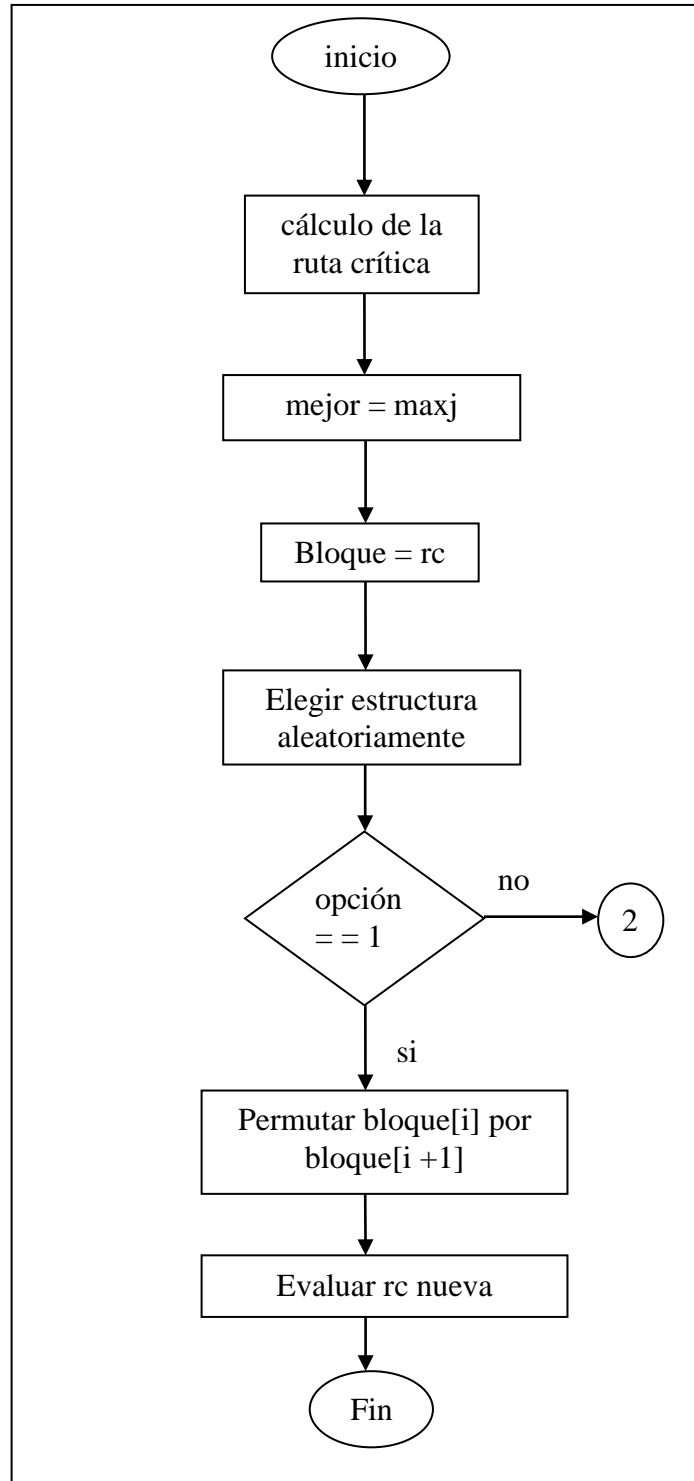


Figura 12. Diagrama de flujo de la estructura de vecindad de la sección correspondiente a la selección del movimiento de vecindad de N1.

BIBLIOGRAFÍA

Referencias bibliográficas

En este apartado se presentan todas las referencias bibliográficas utilizadas para el desarrollo del proyecto.

Bibliografía.

[Blazewicz et al,1996] Jacek Blazewicz , Wolfgang Domschke , Erwin Pesch, The job shop scheduling problem: Conventional and new solution techniques, European Journal of Operational Research 93 (1996) 1-33, October 1995.

[Cruz et al, 2004] Marco Antonio Cruz-Chávez, Juan Frausto-Solís, Fernando Ramos-Quintana, The Problem of Using the Calculation of the Critical Path to Solver Instances of the Job Shop Scheduling Problem, International Journal of Computational Intelligence, ISSN(p): 1304-4508, ISSN(e): 1304-2386, Vol. 1, No. 4, pp. 334-337, 2004.

[Cruz et al, 2007] Marco Antonio Cruz-Chávez, Martín G. Martínez-Rangel, J. A. Hernández, José Crispín Zavala-Díaz, Ocotlán Díaz-Parra, Scheduling Algorithm for the Job Shop Scheduling Problem, Electronics, Robotics and Automotive Mechanics Conference, CERMA2007, IEEE-Computer Society, ISBN 0-7695-2974-7, pp 336-341, 25-28 September, México, 2007.

[Dell'Amico and Trubian, 1993] Mauro Dell'Amico and Marco Trubian, Applying taboo search to the job-shop scheduling problem, Annals of Operations Research 41(1993)231-252, Politecnico di Milano, 1-20133 Milano, Italy.

[Noronha and Sarma, 1991] S. J. Noronha and V. V. S. Sarma, Senior Member, IEEE, Knowledge-Based Approaches for Scheduling Problems: A Survey, IEEE transactions on knowledge and data engineering, vol. 3, no. 2, june 1991.

[Papadimitriou and Steiglitz, 1998] Papadimitriou, C.H., Steiglitz, K. Combinatorial Optimization, Algorithms and Complexity. Dover Publications, Inc. Mineola, New York. USA ed. 1998.

[Roy and Sussman,64] Roy and Sussman, Les problemes d'ordonnancement avec contraintes disjointives, Note D.S. no 9 bis, SEMA, Paris, France, December 1964.