II Conferencia de Ingeniería de Organización
Vigo, 5-6 Septiembre 2002

Algoritmo de Búsqueda Tabú para la Planificación Coordinada de la Producción y Distribución con Ventanas Temporales*

J. M. García [1], G. Villa[2], M. Calle[3], J. L. Andrade[4]
[1]Ingeniero Informático, Dpto. Organización Industrial y Gestión de Empresas. Escuela Superior de Ingenieros. Universidad de Sevilla. Camino de los Descubrimientos, s.n., 41092 - Sevilla , jmgs@esi.us.es
[2]Ingeniero Industrial. gvilla@esi.us.es
[3]Ingeniero de Organización Industrial. mcalle@esi.us.es
[4]Ingeniero de Telecomunicación. jlap_es_pe@yahoo.es

RESUMEN

Este trabajo analiza el problema de seleccionar y planificar una serie de pedidos para ser procesados en una planta de fabricación e inmediatamente distribuidos a la localización del cliente. Las restricciones a considerar son la capacidad limitada de producción en la planta, un número fijo de vehículos disponibles y una ventana temporal para cada pedido dentro de la cuál debería ser entregado el mismo. El problema se describe haciendo alusión a otros problemas similares estudiados en la literatura. Se presenta un modelo de programación entera que maximiza el beneficio asociado a la planificación de los pedidos. Para la resolución del problema se describe un procedimiento basado en búsqueda tabú y se presentan experimentos realizados sobre una batería de problemas generados aleatoriamente. La comparación de los resultados obtenidos de la búsqueda tabú respecto a los obtenidos a través de un método exacto muestra que el procedimiento descrito encuentra soluciones de buena calidad en un corto tiempo de computación.

1. Introducción.

En este artículo se presenta el problema de planificar un conjunto de pedidos haciendo uso de una flota homogénea de vehículos y asumiendo que los pedidos requieren ser fabricados inmediatamente antes de ser distribuidos. Por ello, los pedidos seleccionados en la planificación tienen que ser fabricados en una planta y desde allí ser distribuidos hasta una localización predeterminada.

Asumimos que los datos asociados a cada pedido son conocidos de antemano. Para la fabricación de los pedidos se dispone de una única planta con capacidad de producción limitada. Consideramos capacidad de producción como el número de pedidos que pueden ser preparados simultáneamente, es decir, la fabricación de un pedido se considera un proceso continuo que requiere una unidad de capacidad durante el tiempo que dure el proceso.
En la etapa de distribución de un pedido se consideran tres fases consecutivas: el envío del pedido, la descarga del mismo y la vuelta del vehículo a la planta. Cada vehículo podría transportar cualquier pedido, pero no más de un pedido en un mismo viaje. También se asume que el tamaño del pedido es menor que la capacidad del vehículo. Por todo ello, la fase de distribución de un pedido puede considerarse como un único proceso, que se realiza sin interrupción, y que comienza inmediatamente después del fin de la fase de producción del pedido. Además, como todos los tiempos de proceso (producción y distribución) son

* Este trabajo se deriva de la participación de sus autores en un proyecto de investigación financiado por CICYT con referencia DPI2000-0567.

conocidos, la ventana temporal de entrega puede ser trasladada a una ventana temporal de comienzo de la actividad asociada con el pedido (figura 1).

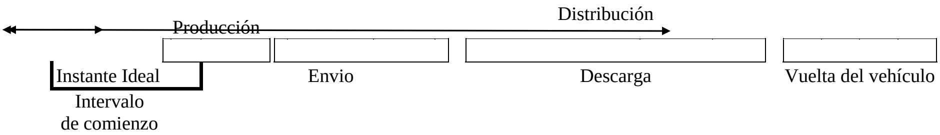

Figura 1. Actividad de un pedido

Este problema se presenta frecuentemente en entornos donde la fase de distribución está conectada a la de producción debido a la ausencia de inventarios del producto final. Estos entornos suelen envolver productos con carácter perecedero. Como ejemplo podríamos mencionar la fabricación y distribución de hormigón. En ese proceso, los materiales que componen la mezcla de hormigón son cargados en el vehículo, y éste es enviado sin interrupción al cliente, debido a que no se dispone de un margen temporal excesivo antes de la solidificación de la mezcla. Situaciones similares aparecen en determinados servicios de reparto de comida.

Para considerar la relevancia del carácter perecedero de esta clase de productos, un instante ideal de entrega se asume dentro de la ventana temporal.

Como en el problema se ha planteado una situación en la que la planta posee capacidad limitada y existe un número finito de vehículos para el envío, podría ocurrir que no fuera admisible atender todas las peticiones dentro de sus ventanas temporales. Por ello y debido a que en el problema se impone que los pedidos deben servirse dentro de su ventana temporal, se considerará como objetivo en el problema la maximización del valor de los pedidos servidos, asumiendo que no todos los pedidos tienen por qué que ser atendidos y que la entrega de un pedido en un instante distinto al ideal supone un decremento del valor del pedido directamente proporcional a la desviación sobre dicho instante, considerando para ello índices correctores sobre el valor del pedido para el caso de adelanto sobre el instante ideal y para el caso de retraso.

En términos de la teoría de secuenciación de trabajos, la planificación coordinada de producción y distribución de pedidos con ventanas temporales envuelve un problema de taller de flujo con dos estaciones formadas por máquinas paralelas, asumiendo *no espera* entre procesos y con diferentes instantes de entrega para los trabajos. La primera estación correspondería con la planta de fabricación, la cuál está compuesta de un número de máquinas idénticas igual a la capacidad de la planta. La segunda estación está compuesta de un número de idénticas máquinas igual al número de vehículos. Cada trabajo correspondería con la producción y distribución de cada pedido.

Los problemas de taller de flujo con máquinas paralelas (FSMP), también llamados líneas de flujo flexible, envuelven la secuenciación de un conjunto de trabajos en un conjunto de estaciones de procesado. Todos los trabajos deben ser procesados sobre cada estación en el mismo orden de estaciones. En ellas, un trabajo puede ser procesado sobre cualquiera de las máquinas. Cuando se asume no espera en el proceso, los trabajos deben procesarse desde la primera hasta la última estación sin existir interrupciones o pausas entre estaciones. FSMP con no espera en proceso ha sido estudiado por diversos autores (ver [1] y [2]). Tanto en FSMP como en FSMP con no espera en proceso, los objetivos considerados han sido siempre los de satisfacer alguna medida de comportamiento que envuelva el procesamiento de todos los trabajos. En la mayoría de los casos, el objetivos es minimizar el instante final de procesamiento de todos los trabajos (makespan). Como un caso diferente, en [3] se estudia el problema FMSP con dos estaciones y no espera en proceso y cuyo objetivo es la maximización del conjunto de pedidos procesados. Sin embargo, no se consideran fechas de entrega para los trabajos. Cuando se consideran fechas de entrega y pesos para trabajos, también se asume que cualquier trabajo puede ser procesado en cualquier instante de tiempo, estableciéndose objetivos como los de minimizar el peso de los trabajos retrasados o similares medidas de comportamiento.

En este artículo presentamos un problema de planificación donde el objetivo es encontrar el subconjunto de trabajos con máximo valor total de forma que esos trabajos puedan ser completados dentro de sus ventanas temporales. Lo que hace a este problema diferente de otros problemas de planificación de trabajos está en que los pedidos servidos deben satisfacer los requerimientos de tiempo impuestos por el cliente. Los problemas de planificación que consideran esta característica son el problema de la planificación de trabajos fijos (FSP) (ver [4], [5]), y el problema de la planificación de trabajos variables (VSP) ([5] y [6]). Sin embargo, estos problemas consideran una única estación de máquinas para el procesamiento de los trabajos, por lo que se ajustarían al caso particular de disponer de capacidad ilimitada de producción, o bien, número ilimitado de vehículos.

El resto del trabajo esta organizado como sigue. En la sección 2 se describe la formulación del problema. En la sección 3 se propone una búsqueda tabú para resolver el problema. La experimentación con una batería de problemas generada aleatoriamente se muestra en la sección 4. Finalmente se extraen las conclusiones del trabajo.

2. Definición del problema y formulación.

Desde una perspectiva asociada a problemas de planificación de trabajos, el problema puede ser definido como sigue: Considérese el problema de encontrar una planificación óptima para un conjunto de n trabajos (pedidos), denotado por $N = \{1,2,...,n\}$, sobre un taller de flujo flexible con no espera en proceso. El taller consiste de dos centros o estaciones de máquinas (fase de producción y fase de distribución) donde el centro de producción está equipado con $C\geq 1$ idénticos recursos o máquinas, que corresponden con la capacidad de la planta. Por otra parte, el centro 2 o centro de distribución posee v máquinas idénticas que se corresponden con la flota de vehículos. Se asocia con cada trabajo una ventana temporal $[a_i,b_i]$ para el comienzo de la actividad asociada con el trabajo, dentro de la cuál se incluye un instante ideal de comienzo s_i. Cada trabajo i tiene un valor positivo w_i reflejando el beneficio obtenido por realizar el trabajo. Este valor w_i asume que el comienzo del trabajo se produce en su instante s_i. Si el instante de comienzo del trabajo fuera anterior (o posterior) a s_i un índice corrector w_i^- (o w_i^+) se aplicaría sobre la desviación para penalizar el beneficio obtenido con el mismo.

Los trabajos seleccionados deben ser procesados de forma continua desde su inicio en el centro de producción hasta su finalización en el centro de distribución, sin ninguna interrupción en los centros ni entre ellos. De ese modo, el trabajo i, $i\in N$, consiste de una secuencia de 2 operaciones, cada una de ellas correspondiente al procesamiento del trabajo i durante un tiempo de proceso tp_i en el centro de producción y td_i en el centro de distribución.

La arquitectura del sistema se muestra en la figura 2.

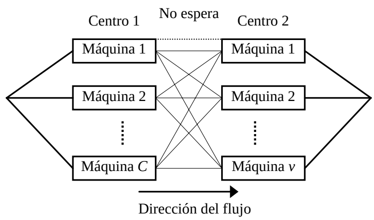

Figura 2. Arquitectura del sistema

Para modelar el problema consideramos la formulación descrita en [5] para la planificación de trabajos variables. Introducimos una escala de tiempo discreto, dividiendo el tiempo en T periodos o instantes, con j=1...T. Dado que a_i representa el instante mas temprano de comienzo y que $d_i = b_i + tp_i$ representa el último instante posible de finalización de la fase de producción, entonces el único conjunto de instantes en los cuales una porción del trabajo podría ser procesada en el centro de producción sería $P_i=\{a_i,...,d_{i-1}\}$. Por otra parte, D_i denota el conjunto de periodos de tiempo sobre el horizonte de planificación en los cuáles el trabajo i podría ser procesado en el centro de distribución. Además, el conjunto I_i recoge los instantes comunes entre P_i y D_i para el trabajo i.

Sea J_j^P el conjunto de trabajos cuya fase de producción podría producirse en el periodo j. De la misma forma, J_j^D define el conjunto de trabajos con fase de distribución asignable al periodo j. Estos conjuntos son usados para asegurar que en cualquier instante, el número total de trabajos asignados en los centros de producción y distribución no excedan del número de recursos disponibles.

Nuestro modelo incorpora varios tipos de variables de decisión relacionadas con el tiempo de proceso de cada trabajo:
y_{ij} =1 si el trabajo i comienza en el instante j: 0 en otro caso

u_{ij} =1 si el trabajo i es procesado en el centro de producción en el instante j: 0 en otro caso

v_{ij} =1 si el trabajo i es procesado en el centro de distribución en el instante j: 0 en otro caso

y_{ij} es usado para marcar el comienzo del procesamiento del trabajo i. u_{ij} y v_{ij} denotan un instante de procesamiento del trabajo i en el centro de producción y distribución, respectivamente. La suma de estas variables u_{ij} sobre $j \in P_i$ debe ser igual al tiempo de producción tp_i. De la misma forma, la suma de las variables v_{ij} sobre $j \in D_i$ debe ser igual al tiempo de distribución td_i. Para determinar los trabajos completados definimos las siguientes variables de decisión: x_i=1 si el trabajo i es procesado; 0 en otro caso.

Obviamente, si x_i=0, las variables x_{ij}, u_{ij} y v_{ij} tomarán valor 0 para cualquier periodo j. Finalmente, definimos las variables n^+_i y n^-_i que recogen la desviación sobre el instante ideal de comienzo de los trabajos seleccionados. Si el trabajo i comienza sobre su instante ideal, ambas variables tomarán valor nulo. Para maximizar el valor total de los trabajos procesados, proponemos el siguiente modelo de programación lineal entera:

$$Max \sum_{i=1}^{n} w_i x_i - w_i^- n_i^- - w_i^+ n_i^+$$

$$s.a.$$

$$\sum_{j \in P_i} u_{ij} = tp_i x_i \quad i = 1 \ldots n \qquad (1)$$

$$\sum_{j \in P_i} v_{ij} = td_i x_i \quad i = 1 \ldots n \qquad (2)$$

$$tp_i u_{ij} - tp_i u_{ij+1} + \sum_{k=j+2}^{f_i} u_{ik} \le tp_i \quad i = 1 \ldots n; \ \forall j \in P_i \qquad (3)$$

$$td_i v_{ij} - td_i v_{ij+1} + \sum_{k=j+2}^{f_i} v_{ik} \le td_i \quad i = 1 \ldots n; \ \forall j \in D_i \qquad (4)$$

$$(u_{ij} + v_{ij}) - u_{ij-1} = y_{ij} \quad i = 1 \ldots n; \ \forall j \in P_i \quad j \ne a_i \qquad (5)$$

$$\sum_{i \in J_j^P} u_{ij} \le C \quad j = 1 \ldots T \qquad (6)$$

$$\sum_{i \in J_j^D} v_{ij} \le V \quad j = 1 \ldots T \qquad (7)$$

$$u_{ij} + v_{ij} \le 1 \quad i = 1 \ldots n; \forall j \in I_i \qquad (8)$$

$$u_{ij} - (u_{ij+1} + v_{ij+1}) \le 0 \quad i = 1 \ldots n; \forall j \in I_i \qquad (9)$$

$$(\sum_{i \in P_i} j y_{ij} - s_i x_i) + n_i^- - n_i^+ = 0 \quad i = 1 \ldots n \qquad (10)$$

$$u_{ij} = 0 \ \ \forall j \notin P_i; v_{ij} = 0 \ \ \forall j \notin D_i \quad i = 1 \ldots n \qquad (11)$$

$$x_i, y_{ij}, u_{ij}, v_{ij} \in \{0,1\} \quad n_i^+, n_i^- \ge 0$$

Restricciones (1) y (2) aseguran que cada trabajo es asignado exactamente a un número de periodos igual a su tiempo de proceso en el centro de producción y distribución, respectivamente. Restricciones (3) y (4) imponen que cada trabajo es asignado a un conjunto adyacente de periodos de tiempo, tanto en la fase de producción como en la de distribución. Restricciones (5) definen el instante de comienzo de un trabajo. Restricciones (6) aseguran que no mas de *C* trabajos en la fase de producción son asignados a cualquier periodo. De la misma forma, restricciones (7) permiten a lo sumo *V* trabajos procesándose en la fase de distribución durante cualquier instante de tiempo. Juntas, las restricciones (8) y (9) fuerzan que la etapa de distribución de cada trabajo comience, sin retraso, justo después del último periodo en el que finaliza la fase de producción del trabajo. En (10) se define, para los trabajos procesados actualmente, el número de periodos en los que se adelanta o retrasa el trabajo sobre su instante ideal de comienzo s_i. Finalmente, las restricciones (11) definen los periodos de tiempo en los que cada trabajo podría ser procesado.

3. Algoritmo de búsqueda tabú propuesto.

En esta sección se describe la búsqueda tabú propuesta para resolver el problema PDPTW. Los parámetros del algoritmo son descritos a continuación:

Solución Inicial: Para obtener una solución inicial, consideramos un algoritmo GRASP basado en un método GRASP descrito en [7] para el mismo problema considerando instantes fijos de entrega para los pedidos. En este caso, en cada iteración del algoritmo asumimos que el instante de inicio de un pedido, que no pertenece a la solución que está siendo construida, es el instante dentro de su ventana temporal que proporciona el mejor valor admisible.

Movimientos y estructura de la vecindad: Dada una solución *s* sea N(*s*) el conjunto de todas las soluciones admisibles que pueden ser obtenidas desde *s* usando uno de los siguientes movimientos:

Cambio: Una solución vecina de *s* se obtiene por un procedimiento de cambio donde un trabajo *i* perteneciente al conjunto solución es reemplazado por otro/s trabajo/s que no está/n en el conjunto solución. Denotamos por *S* el conjunto que contiene los trabajos que pertenecen al conjunto solución actual. Dado un trabajo $i \in S$, consideramos todos los trabajos *j* con $j \notin S$ cuya fase de producción o de distribución coincide en algún instante con la del trabajo *i*. Usemos H_i para denotar el conjunto de esos trabajos. Los trabajos pertenecientes a H_i son primero ordenados por valor y tentativamente introducidos en el conjunto solución *S* para chequear si las restricciones del problema siguen cumpliéndose. En el proceso de evaluación se examina cualquier posible instante de comienzo del trabajo. La figura 1 ilustra este movimiento para el caso en que reemplazamos de la solución el trabajo B.

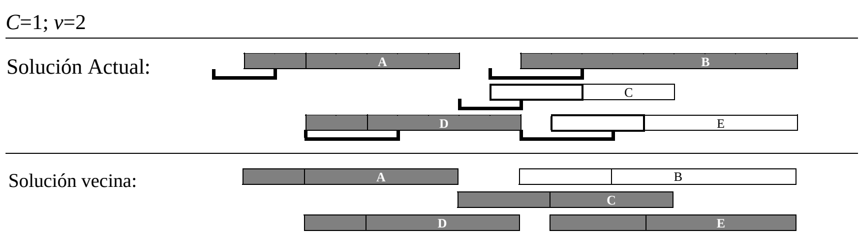

Figura 3: Movimiento de cambio para el trabajo B.

Quitar: Cuando el conjunto H_i es vacío el movimiento consiste únicamente en sacar de la solución el trabajo *i*. Este tipo de movimiento lleva, lógicamente, a soluciones peores. Sin embargo, llega a ser útil como herramienta de diversificación. Este movimiento es únicamente utilizado cuando los trabajos de la solución que dan lugar a un movimiento de cambio son tabú. Como ilustración, si en el ejemplo de la figura 1, reemplazar los trabajos B y D son considerados tabú en una determinada iteración, el único movimiento posible es quitar el trabajo A.

Lista tabú: El tamaño de la lista tabú es un parámetro muy importante en un algoritmo de búsqueda tabú. La lista tabú puede ser fija o variable. Glover, [8], [9], [10], sugiere el uso de listas tabú variables para obtener mejores resultados. Nosotros intentamos una estrategia donde el tamaño de la lista tabú se varía para ajustar el número de iteraciones en las que se acepta un movimiento de quitar un trabajo. Los experimentos preliminares mostraron que este número no debe exceder el número de trabajos del problema. Así, los mejores resultados se obtuvieron para un tamaño igual al número de pedidos en la solución inicial dividida por dos.

Criterio de aspiración: Para el criterio de aspiración usamos la forma más clásica en la que un movimiento tabú es aceptado si produce una solución mejor que la mejor solución obtenida hasta ese momento.

Criterio de parada: El algoritmo se detiene después de un número máximo de iteraciones o cuando se encuadra la solución óptima del problema. Se tomó un máximo de 5000 iteraciones.

4. Resultados computacionales.

La primera fase de nuestros experimentos envolvió la construcción de la batería de problemas. Para ello, se usó como parámetro principal el promedio de solapamiento de los pedidos en las fases de producción (PSP) y distribución (PSD). De ese modo, los valores considerados para PSP y PSD estuvieron siempre contenidos en los intervalos (1.5,1.6) y (5,6) respectivamente.

Los tamaños de los problemas usados fueron 20, 25, 30, 40 y 50 pedidos. Diez instancias fueron generadas aleatoriamente con un tamaño entre 1 y 5 periodos. El horizonte de tiempo en los problemas ha sido considerado dependiente del número de pedidos del problema, de acuerdo a los siguientes intervalos: [1,55] para 20 pedidos; [1,65] para 25 pedidos; [1,75] para 30 pedidos; [1,95] para 40 pedidos y [1,115] para 50 pedidos. Los valores de los pedidos fueron generados aleatoriamente dentro del intervalo [10,100] y las penalizaciones tanto para retrasos como adelantos fueron seleccionadas aleatoriamente dentro del intervalo [0,2]. Para permitir diferentes niveles con respecto a la capacidad C y al número de vehículos v, los siguientes pares de valores (C,v) fueron considerados en cada problema: (1,2); (2,3).

Para medir el comportamiento del algoritmo, inicialmente resolvimos el conjunto de problemas usando un procedimiento exacto basado en un grafo. Este método está basado en un procedimiento descrito en [7] para el mismo tipo de problema pero sin considerar ventanas temporales, es decir, con instantes fijos de entrega. El método construye un grafo G que colecciona todas las soluciones admisibles del problema a través de un simple método de evaluación de estados admisibles en la secuenciación de los pedidos. El valor del camino máximo desde el nodo de salida a el nodo final en G es la solución óptima para el problema. Con ventanas temporales, la evaluación de todos estados admisibles implica considerar cada periodo en el que podría comenzar cada pedido.

Las tablas 1 y 2 muestran el sumario de resultados usando tanto el método exacto como la búsqueda tabú. Los porcentajes de error han sido computados con respecto a los valores óptimos obtenidos a través del método exacto y teniendo en cuenta también, que en cada tamaño de problema n se hace un promedio sobre las 10 instancias empleadas. Todos los tiempos de ejecución vienen expresados en segundos sobre un Intel Pentium III 850 MHz.

n/C/v	Promedio Error (%)	Número de Soluciones óptimas encontradas	Tiempo de computación total de BT	Promedio del nº de iteraciones en BT	Tiempo hasta la mejor solución (BT)	Tiempo de computación del procedimiento exacto
20/1/2	0.00	10	7.5	49.7	0.3	73.1
20/2/3	0.06	9	11.2	1451.3	4.5	1496.0
25/1/2	0.32	9	9.7	494.5	2.1	278.2
25/2/3	0.24	9	13.8	1270.7	4.7	1957.0
30/1/2	0.41	8	11.1	1908.5	4.5	194.1
30/2/3	0.30	7	16.6	1144.3	7.4	6898.0
40/1/2	0.21	7	14.1	2270.0	7.9	97.7
40/2/3	0.50	1	20.0	2032.5	17.9	12966.0
50/1/2	0.12	5	17.0	1230.5	11.3	178.5
50/2/3	0.62	1	27.1	2258.0	21.7	13314.0

Tabla 1. Sumario de resultados

La búsqueda tabú implementada encontró soluciones óptimas en 66 de los 100 problemas resueltos. Aunque el error aumenta con C y v, nunca excede del 1%. Con respecto a los tiempos de computación, el método exacto tomó significativamente más tiempo que la metaheurística implementada.

4. Conclusiones.

En este trabajo hemos estudiado un tipo de problema de planificación para la producción y distribución sin pausas entre procesos y con ventanas temporales en las entregas. Una heurística basada en búsqueda tabú se ha implementado para la resolución del problema. La calidad de las soluciones proporcionadas por la búsqueda tabú se comparó con las soluciones óptimas de los problemas, las cuales fueron proporcionadas por un método exacto de solución basado en la construcción de un grafo de estados admisibles. Los resultados computacionales indican que la heurística encuentra soluciones de muy buena calidad en un breve tiempo de computación.

Referencias.

[1] Hall N.G. and Sriskandarajah C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research*, 44, 510-525.

[2] Sriskandarajah C. (1993). Performance of scheduling algorithms for no-wait flowshops with parallel machines. European Journal of Operations Research, 70, 365-378.

[3] Ramudhin A. and Ratliff H.D. (1995). Generating daily production schedules in process industries. IIE Transactions, 27, 646-656.

[4] Kroon L.G., Salomon M. and Van Wassenhove L. N. (1995). Exact and approximation algorithms for the operational fixed interval scheduling problem. European Journal of Operational Research, 82, 190-205.

[5] Gertsbakh I. and Stern H. (1978). Minimal Resources for Fixed and Variable Job Schedules. Operations Research, 26 (1), 68-85.

[6] Gabrel V. (1995). Scheduling jobs within time windows on identical parallel machines: New model and algorithms. European Journal of Operations Research, 83, 320-329.

[7] Garcia J. M., Smith K., Lozano S. and Guerrero F. (2001). A Comparison of GRASP and an Exact Method for Solving a Production and Delivery Scheduling Problem, Proceedings of the International workshop on Hybrid Intelligent Systems HIS'2001, Adelaide, Australia.

[8] Glover F. (1989). Tabu search. Part I. ORSA Journal on Computing, 1, 190-206.

[9] Glover F. (1990). Tabu search. Part II. ORSA Journal on Computing, 2, 4-32.

[10] Glover F., Taillard E. and Werra D. (1993). A user’s guide to tabu search. Annals of Operations Research, 41, 3-28.