

Planificación Coordinada de la Producción y Distribución de Pedidos desde Múltiples Plantas*

J. M. García¹, M. Calle², G. Villa³, J. L. Andrade⁴

¹Ingeniero 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

²Ingeniero de Organización Industrial. mcalle@esi.us.es

³Ingeniero Industrial. gvilla@esi.us.es

⁴Ingeniero de Telecomunicación. jlap_es_pe@yahoo.es

RESUMEN

Este trabajo se ocupa del problema asociado a la planificación conjunta de la fabricación y distribución de pedidos para el caso de múltiples plantas de producción. El problema busca la selección de pedidos para ser procesados desde las plantas de producción existentes, e inmediatamente distribuidos a la localización del cliente. Se presenta un modelo de programación entera que maximiza el beneficio de los pedidos servidos. Tanto un método exacto como un procedimiento heurístico basado en un problema de flujo a coste mínimo son descritos como herramientas de resolución.

1. Introducción.

En ciertos entornos industriales el abastecimiento de productos debe estar acorde con la demanda debido a la ausencia de inventarios del producto. Esta característica suele estar motivada por el carácter perecedero de los productos fabricados. Ejemplos se pueden encontrar en la producción y distribución de hormigón y en el reparto y preparación de comida rápida. Debido a la naturaleza de estos productos, los pedidos que se hagan del producto necesitan ser fabricados inmediatamente antes de ser repartidos al lugar de destino.

Este trabajo considera el problema de planificar un conjunto dado de pedidos, de forma que los pedidos seleccionados para ser servidos deben ser preparados en una cualquiera de las plantas de producción existentes, e inmediatamente distribuidos al lugar del cliente. Existen un conjunto de m plantas de producción $K=\{1\dots m\}$, cada una con una capacidad de producción limitada c_k . Consideramos capacidad de producción como el número de pedidos que pueden ser preparados simultáneamente, es decir, la producción de un pedido se considera un proceso continuo que requiere una unidad de capacidad durante su tiempo de proceso. Si la planta k tiene c_k unidades de capacidad, entonces a lo sumo c_k pedidos pueden ser preparados simultáneamente. Además, cada pedido i puede únicamente ser preparado en un subconjunto K_i del conjunto de plantas K .

Existe un número fijo de vehículos de idénticas características. Una vez que un pedido se prepara en la planta, se distribuye sin esperas al lugar de destino. Asumimos que el tamaño de cada pedido es inferior a la capacidad de un vehículo y que solo un pedido puede ser distribuido por viaje que realice un vehículo. Cuando un pedido es descargado en el destino, el

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

vehículo retorna a cualquiera de las plantas existentes y queda disponible para el reparto de otro pedido. Inicialmente, cada vehículo se encuentra en una planta específica. Consideraremos v_k como el número de vehículos inicialmente en la planta k .

El número de pedidos, sus tiempos de proceso en planta y los tiempos de viaje desde la localizaciones de destino a cualquiera de las plantas son datos conocidos y fijos. De ese modo, con cada pedido i , del conjunto de pedidos $O=\{1\dots n\}$, se asocia una fecha de entrega e_i , un tiempo desde cada planta k a su localización ti_{ik} , un tiempo de viaje desde esa localización hasta las plantas tr_{ik} , un tiempo de proceso o producción tp_i independiente de la planta, y un tiempo de descarga tu_i .

Además, cada pedido tiene un valor positivo w_i reflejando el beneficio asociado con servirlo. Con respecto a los costes del problema, por cada unidad de tiempo asociaremos una unidad de coste. Los costes de producción son los mismos para todas las plantas y por tanto no se consideran.

Debido a que tenemos una capacidad limitada en cada planta así como un número finito de vehículos, podría suceder que no fuera admisible servir todos los pedidos solicitados. Por ello, el objetivo en el problema será maximizar el beneficio asociado a la selección de pedidos planificada, considerando tanto el valor de los pedidos servidos como los costes de transporte a que han dado lugar. La figura 1 muestra el proceso gráfico asociado a la actividad de un pedido.

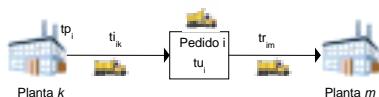


Figura 1. Representación gráfica

El trabajo está organizado como sigue. En la sección 2 formulamos el problema como un problema de programación lineal entera. En la sección 3 consideramos el caso donde la capacidad de producción de cada planta es mayor o igual al número de vehículos disponibles. Para este caso, presentamos un método óptimo que puede ser considerado como un procedimiento para obtener una cota superior de la solución óptima para el caso general. En la sección 4 se propone un método basado en grafos para obtener la solución exacta del problema general. Sin embargo, cuando los problemas son lo suficientemente grandes se descarta el uso de este método. Por ello, una heurística es también propuesta. Resultados computacionales se describen en la sección 5 y, finalmente, en la sección 6 se muestran las conclusiones.

2. Formulación.

Para formular el problema introducimos una escala de tiempo discreto donde se divide el tiempo en T periodos, con $t = 1\dots T$. Sea $l_{ik} = s_{ik} + tp_i$ el tiempo de comienzo de la distribución del pedido i si fuera servido desde la planta k , y sea $f_{ik} = e_i + tu_i + tr_{ik}$ el tiempo de llegada del vehículo a la planta k , una vez servido el pedido.

Para cada pedido $i \in O$ y cada planta $k \in K_i$, en todo instante de tiempo $t \in [s_{ik}, l_{ik} - 1]$ definimos $S(i,t,k) = \{j \in O: s_{jk} \leq t \text{ \& } l_{jk} > t\}$ como el conjunto de pedidos cuyo fase de producción en la planta k se solapa con la del pedido i en esa misma planta y en el instante t . Sea d_{ik} el coste de transportar el pedido i desde la planta k y d'_{ik} el coste de la vuelta del vehículo a la planta desde la localización del pedido i .

Nuestro modelo incorpora las siguientes variables de decisión:

$x_{ij} = 1$ si el pedido i se procesa en la planta k ; 0 en otro caso. ($i \in O$ y $k \in K_i$)

$z_{ik} = 1$ si el vehículo que sirve el pedido i vuelve a la planta k ; 0 en otro caso. ($i \in O$ y $k \in K$)

$y_{t,k}$ = número de vehículos disponibles en el instante t_r en la planta k . El conjunto $\{t_r: r = 1, \dots, R_k\} = \{l_{ik}, \forall i: k \in K_i\} \cup \{f_{ik}, \forall i\}$, y $t_{r-1} \leq t_r$ para $r = 1 \dots R_k$. En cada una de las plantas, el instante t_o indica el instante inicial sin actividad. El modelo resultante es:

$$\begin{aligned} & \text{Max} \sum_{i=1}^n \sum_{k \in K_i} w_i x_{ik} - \left(\sum_{i=1}^n \sum_{k \in K_i} d_{ik} x_{ik} + \sum_{i=1}^n \sum_{k=1}^K d'_{ik} z_{ik} \right) \\ & \text{subject to} \\ & \sum_{k \in K_i} x_{ik} \leq 1 \quad \forall i \in O \quad (1) \\ & \sum_{k \in K_i} x_{ik} - \sum_{k=1}^K z_{ik} = 0 \quad \forall i \in O \quad (2) \\ & x_{ik} \leq y_{t,k} \quad t_r = l_{ik} \quad (3) \\ & y_{t_0,k} = v_k \quad \forall k \in K \quad (4) \\ & y_{t_{r+1},k} = y_{t_r,k} - x_{ik} \quad \forall i \in O: t_r = l_{ik} \quad (5) \\ & y_{t_r,k} = y_{t_{r-1},k} + z_{ik} \quad \forall i \in O: t_r = f_{ik} \quad (6) \\ & \sum_{j \in S(i,t,k)} x_{jk} \leq c_k \quad \forall i \in O, \forall k \in K_i, \forall t \in [s_{ik}, l_{ik} - 1] \quad (7) \\ & x_{ik}, z_{ik}, y_{t,k} \in \{0,1\} \end{aligned}$$

La función objetivo maximiza el beneficio obtenido con los pedidos satisfechos. Las restricciones (1) imponen que un pedido simplemente puede ser procesado en, a lo sumo, una planta. Las restricciones (2) aseguran que después de que un pedido sea servido (descargado), el vehículo vuelve a una planta. Las restricciones (3) imponen que un pedido podría ser procesado en la planta k si hay vehículos disponibles en el instante l_{ik} . Las restricciones (4) establecen el número inicial de vehículos en cada planta. Las restricciones (5) y (6) computan el número de vehículos disponibles en cada planta en cada instante t_r . Finalmente, las restricciones (7) imponen que no más de c_k pedidos sean asignados a cualquier periodo de tiempo en cada planta k .

3. Número de vehículos menor o igual a la capacidad de producción de cada planta.

Podemos definir el número total de vehículos como $V = \sum_{k=1}^m v_k$

Si $V \leq c_k, \forall k \in K$, el problema cobra un cierto interés puesto que este caso sería el problema de encontrar un subconjunto de trabajos con beneficio máximo y que puedan ser servidos por los vehículos disponibles. Como V no excede la capacidad de producción en cada planta, eso significa que existe capacidad suficiente en cada planta para procesar todos los pedidos que pudieran estar siendo distribuidos en cualquier instante. Este problema puede ser resuelto en tiempo polinomial por un algoritmo de flujo a coste mínimo (ver [1]). La construcción del grafo directo G que usaremos para resolver el problema puede ser descrito como sigue:

Por cada planta existirá un nodo en G . Representamos esos nodos con la nomenclatura $\{n_1, \dots, n_k\}$. Para cada pedido i , creamos un subgrafo G_i con la siguiente estructura: Nodos $n_{i,ok}, k=1 \dots m$, definen cada una de las plantas donde el pedido i pudiera ser procesado, con independencia de que $k \notin K_i$. Nodos n_{is} y n_{ie} definen el comienzo y fin de la descarga del

pedido. Los nodos n_{rk}^i , $k=1..m$, representan las posibles plantas de retorno para el vehículo que sirve el pedido i . Los tiempos desde cada planta al lugar de destino del pedido i se representan como costes en los arcos que unen los nodos asociados a las plantas, n_{ok}^i , con los nodos de comienzo de descarga, n_{is} . Los tiempos de retorno son también representados en G_i de forma similar. El valor del pedido i se representa como un coste negativo $-w_i$ en el arco que define la descarga. Como un pedido es procesado a lo sumo una vez, la capacidad para cada arco en G_i es 1. Sin embargo, cuando una planta k no pertenece a K_i , la capacidad del arco que representa este viaje es cero. Además, para cada nodo planta n_k existe un arco desde n_k a n_{ok}^i , $i = 1..n$, con capacidad de uno y coste cero.

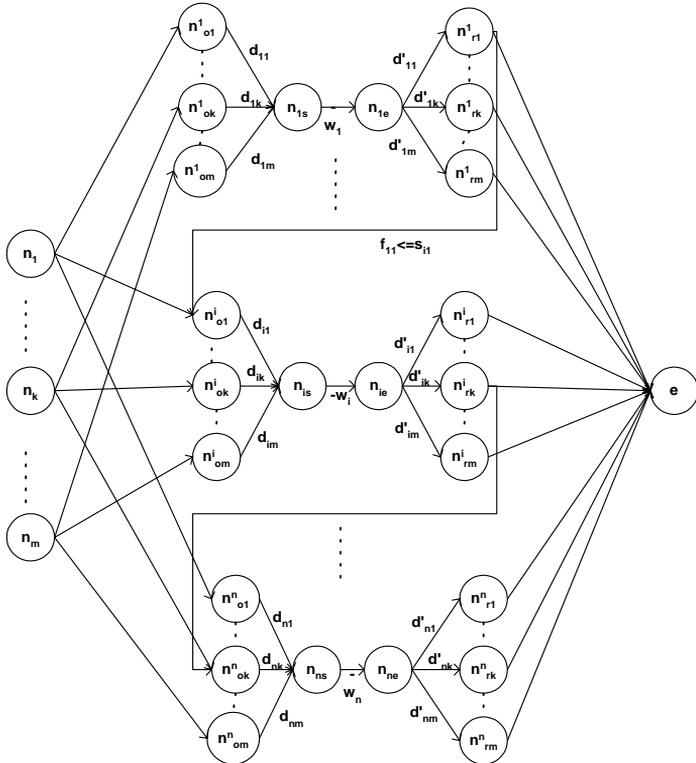


Figura 2. Gráfico genérico G

Para permitir que un vehículo que retorna a una planta pueda servir a otros pedidos desde esa planta, se aplica la siguiente regla para crear nuevos arcos en G que reflejen esta posibilidad:
 { Crear arco desde n_{rk}^i a n_{dk}^j si $f_{ik} \leq s_{jk}$ }

El número de arcos que pueden ser creados con esta regla es menor o igual que $m(n^2-n)/2$. Finalmente, se conectan todos los nodos n_{rk}^i , $i = 1..n$, $k = 1..m$, a un nodo final e . Estos arcos no tienen coste y pueden almacenar una unidad de flujo. Un grafo genérico G se muestra en la figura 2. El número de nodos en G es igual a $m + 2m + 2n$. El número de arcos es menor o igual a $3mn + n + m(n^2-n)/2$. Si inyectamos v_k (es decir, el número de vehículos iniciales en la planta k) unidades de flujo en el nodo n_k , $k = 1..n$, el camino con coste mínimo en G proporcionará los pedidos a procesar. Obviamente, un pedido i es procesado si una unidad de flujo circula a través del subgrafo G_i .

Este procedimiento puede ser usado como método de obtención de una cota superior de la solución óptima para problemas en los que no existan las limitaciones impuestas en este caso particular, es decir, donde no se cumpla que $V \leq c_k$.

4. Dos enfoques para resolver el problema.

En esta sección proponemos dos estrategias para resolver el problema en el caso general sin limitaciones. La primera estrategia usa un método exacto para obtener soluciones óptimas. La razón principal para desarrollar este método es que sus resultados pueden guiar las conclusiones sobre el método heurístico descrito en la sección 4.2.

4.1 Un método exacto basado en un grafo.

El enfoque que presentamos está basado en un método efectivo para el mismo pero con una sola planta de producción (ver [2]). El método construye un grafo G que colecciona todas las soluciones admisibles del problema a través de un método de evaluación de estados admisibles en la planificación de los pedidos. El camino de peso máximo desde el nodo de salida al nodo de final en G es la solución óptima del problema.

El proceso de obtención del grafo G se describe en el siguiente algoritmo:

Sea n_o el estado de comienzo y n_e el estado final

Sea N_{ik} el conjunto de creados al comienzo del procesamiento del pedido i en la planta k

Para cada pedido i y cada planta $k \in K_i$ (ordenado por s_{ik})

Para cada estado $n_j \in N_{rs}/s_{rs} < s_{ik}$

Sea $n'_j = \{\text{pedidos} \in n_j \text{ que continúan procesándose en el instante } s_{ik} \text{ en la planta } k\}$

Sea y_k el número de vehículos disponibles en el instante t_{ik} en la planta k , $k=1 \dots m$

Si máximo grado de solapamiento en la planta k de $\{n'_j \cup i\} \leq c_k$ & $y_k > 0$

Para cada planta $k' \in K$

Crear Estado $n_{new} = \{n'_j \cup i\}$ en N_{ik} finalizando el vehículo en k'

Crear Arco desde n_j a n_{new}

Siguiente

Fin Si

Siguiente

Siguiente

Unir todos los estados que sean terminales al estado final n_f

El número de nodos y arcos en G es de orden $O(2^{nk})$, es decir, crece de forma exponencial con el tamaño del problema.

4.2 Un enfoque heurístico.

La heurística que ahora presentamos se basa en la idea descrita para el problema en la sección 3. El algoritmo comienza resolviendo un problema de flujo a coste mínimo para el grafo definido en esa sección (sección 3). Sea G^o ese grafo. Una vez que la solución es obtenida para G^o , la admisibilidad de los pedidos seleccionados en cada planta es chequeado. Para esta tarea, haremos uso del lema de *Kroon* [3] sobre el problema de la planificación de trabajos fijos (FSP), con el que se dará una condición necesaria y suficiente para la existencia de un plan admisible para todos los pedidos seleccionados en cada planta:

Sea O_k el conjunto de pedidos seleccionados para ser procesados en la planta k . Un plan admisible que incluya a todos los pedidos de O_k existe si y sólo si el solapamiento máximo de

los pedidos de O_k en la planta k es menor o igual a la capacidad c_k de la planta. Supongamos que los pedidos se procesan en un intervalo de tiempo $[0, T]$. El índice de solapamiento máximo en la planta k se define entonces como sigue: $L_k = \max \{L_{tk}; 0 \leq t \leq T\}$ con $L_{tk} = \{i \in O_k; s_{ik} \leq t \leq l_{ik} - 1\}$

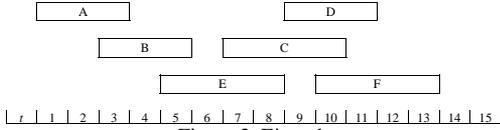


Figura 3. Ejemplo

La Figura 3 presenta un ejemplo de un problema con seis pedidos. En la figura las barras indican el tiempo de producción (tp_i) de los pedidos. En este ejemplo L_k es igual a 3. Es claro que solo si $c_k \geq 3$ podrán procesarse todos los pedidos en la planta.

Si el máximo índice de solapamiento de O_k excede de c_k entonces la solución proporcionada por el problema de flujo a coste mínimo no es admisible. En este caso, la heurística intentará encontrar un subconjunto de pedidos con beneficio máximo que pueda ser procesado con una capacidad de c_k . Este problema es equivalente al problema de maximizar la planificación de trabajos fijos (Max. FSP). Max. FSP ha sido considerado por un número de autores que incluyen a Arking y Silverberg [4], Kroon, Salomón y Van Wassenhove [3] y Gabrel [5], quienes muestran que el problema puede ser resuelto por un algoritmo de flujo a coste mínimo.

La construcción del grafo G_k que usamos en este trabajo es más directo que las construcciones propuestas por esos autores, y puede ser descrita como sigue. El conjunto $R = \{r_p; p = 1 \dots P\}$ se usa para representar todos los tiempos de comienzo de los trabajos que pertenecen a O_k en orden cronológico. Es decir, $R = \{s_{ik}; i \in O_k\}$ and $r_{p-1} < r_p$. El conjunto de nodos del grafo se corresponde uno a uno con el conjunto R , además de un nodo final. Existe un arco desde cada nodo al siguiente con coste cero y capacidad ilimitada. Por otra parte, existen arcos desde cada nodo al nodo correspondiente al primer pedido que podría ser producido en la planta, una vez que hubiera finalizado la producción del pedido representado por el nodo que es origen del arco. Estos arcos poseen capacidad para contener una unidad de flujo y el coste es igual a menos el beneficio por producir el pedido en la planta k , es decir, un coste igual a $-(w_i - d_{ik})$. En el primer nodo del grafo o nodo origen se inyectan c_k unidades de flujo, las cuales deben alcanzar el nodo final. Como ejemplo, en la figura 4 se muestra el grafo correspondiente al ejemplo de la figura 3.

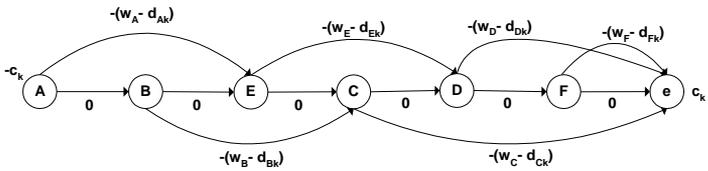


Figura 4. Grafo asociado al ejemplo de la figura 3

Una vez obtenida la solución óptima de este problema de flujo, denotemos por E_k el conjunto de pedidos de O_k que no han sido seleccionados. Para cada pedido $j \in E_k$, modificamos el grafo G^0 para no permitir que el pedido j pueda ser procesado en la planta k . Para tal fin, tenemos que asignar capacidad cero al arco que une n'_{ok} con n_{js} . Con este nuevo grafo G^0 , todo el

proceso descrito hasta ahora se repite de nuevo hasta encontrar una solución admisible. La figura 5 ilustra el algoritmo en pseudocódigo.

```

Crear grafo G0
Repetir
  Calcular flujo a coste mínimo en G0
  Para cada planta k ∈ K
    Sea Ok = {i ∈ O / flujo del arco (niok, nik) = 1}, k = 1...m
    Crear grafo Gk
    Calcular flujo a coste mínimo en Gk
    Ek = {j ∈ Ok : flujo en el arco (j, posterior(j)) = 0}
    Si Ek = ∅ entonces
      Admisibilidad en k = True
    Si no
      Admisibilidad en k = False
      Capacidad de los arcos (niok, njk) = 0 en G0, j ∈ Ek
    End if
Hasta que Admisibilidad en k = True para ∀ k
  
```

Figura 5. Pseudocódigo del procedimiento heurístico

5. Resultados experimentales.

En esta sección presentamos y analizamos los resultados obtenidos en los experimentos llevados a cabo. El tamaño de los problemas han sido de $n=10$ y $n=20$ pedidos. El horizonte de tiempo de los problemas se consideró dependiente del número de pedidos, de acuerdo a los siguientes intervalos: [1,50] para 10 pedidos y [1,60] para 20 pedidos. Los tiempos de viaje, producción y descarga, así como el valor de los pedidos, fueron obtenidos aleatoriamente dentro de los intervalos mostrados en la tabla 1.

Tiempos de viaje	[4,10]	Tiempos de descarga	[1,2]
Tiempos de Producción	[1,5]	Valor de pedidos	[30,100]

Tabla 1. Intervalos para la generación aleatoria de datos

Para permitir diferentes niveles con respecto al número de plantas m y el número de vehículos V , los siguientes pares de valores fueron considerados para cada valor de n :

- $n=10$: $(m, V) = (2,2); (2,3); (3,2); (3,3)$
- $n=20$: $(m, V) = (2,2); (3,2)$

Fue considerado que cada pedido i podría ser preparado en cualquiera de las plantas. La capacidad de producción c_k para cada planta en todos los problemas fue de 1 unidad. El número de vehículos iniciales en cada planta se obtuvo de forma aleatoria para cada problema. La tabla 2 muestra, para seis instancias, el comportamiento del método exacto con relación al número de estados (nodos), número de arcos y tiempo de computación. Todos los tiempos vienen dados en segundos de CPU sobre un Pentium III a 850 Mhz.

Instancia	n	M	V	Nº de nodos	Nº de arcos	Tiempo computación
1	10	2	2	996	1738	1
2	10	2	3	16874	29178	18
3	10	3	2	15392	27741	7
4	10	3	3	1003295	1835580	813
5	20	2	2	177233	314993	151
6	20	3	2	1389857	2519008	1011

Tabla 2. Comportamiento del método exacto

Respecto al comportamiento de la heurística propuesta, la tabla 3 muestra los resultados obtenidos sobre 10 instancias de cada valor de n , y (m, V) . Los porcentajes de error han sido computados con respecto a la soluciones óptimas obtenidas a través del método exacto.

n	m	V	Porcentaje de Error (%)	Nº de soluciones óptimas encontradas	Promedio del tiempo de Computación	Promedio de pedidos servidos
10	2	2	1.07	9	1	4.9
10	2	3	0.40	9	1	6
10	3	2	0.00	10	1	5.1
10	3	3	0.02	9	1	6.1
20	2	2	0.43	7	2	6.8
20	3	2	0.54	9	2	6.5

Tabla 3. Sumario de resultados

Nótese que:

- La heurística encuentra soluciones óptimas en 53 de los 60 problemas resueltos.
- El tiempo de computación del método heurístico nunca excedió de 2 segundos.

En vista de los resultados obtenidos, podemos concluir que la heurística implementada presenta buenos resultados.

6. Conclusiones.

En este trabajo hemos estudiado un problema de planificación asociado a entornos de producción y distribución coordinada con varias plantas. Hemos considerado un caso particular que puede ser resuelto en tiempo polinomial por un algoritmo de flujo. Además, este método ha sido para construir una heurística para el problema en un caso general. La calidad de la soluciones proporcionada por la heurística se ha comparado empíricamente con las soluciones óptimas de los problemas, obtenidas éstas a través de un método de solución exacta. Resultados computaciones indican que la heurística encuentra rápidamente soluciones óptimas o muy cercanas al óptimo.

Referencias.

- [1] Bertsekas D. and Tseng P. "Relaxation methods for minimum cost for ordinary and generalized network flow problems". *Operations Research*, 36, 93-114. (1988).
- [2] Garcia J. M., Smith K., Lozano S. and Guerrero F. "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. (2001).
- [3] Kroon L.G., Salomon M., Van Wassenhove L. N., "Exact and approximation algorithms for the operational fixed interval scheduling problem," *European Journal of Operational Research* 82, pp. 190-205, (1995).
- [4] Arking E.M., Silverberg E.B., "Scheduling jobs with fixed start and end times," *Discrete Applied Mathematics* 18, pp. 1-8, (1987).
- [5] Gabrel V., "Scheduling jobs within time windows on identical parallel machines: New model and algorithms," *European Journal of Operations Research* 83, pp. 320-329, (1995).