X Congreso de Ingeniería de Organización
Valencia, 7 y 8 de septiembre de 2006

Diseño de subastas para la programación Job Shop

Araúzo Araúzo, José Alberto [1], de Benito Martín Juan José [2], Sanz Angulo Pedro[3], del Olmo Martínez Ricardo[4]
[1] Ingeniero Industrial, ETSII. de la UVA, Paseo del Cauce s/n 47011 Valladolid, arauzo@eis.uva.es
[2] Dr. Ingeniero Industrial, ETSII. de la UVA, Paseo del Cauce s/n 47011 Valladolid, debenito@eis.uva.es.
[3] Dr. Ingeniero Industrial, ETSII. de la UVA, Paseo del Cauce s/n 47011 Valladolid, psangulo@eis.uva.es.
[4] Dr. Ingeniero Industrial, EPS. de la UBU, Avenida Santander s/n 09006 Burgos, rdelolmo@ubu.es.

RESUMEN

La programación detallada de los sistemas de producción tipo "Job Shop" ha sido motivo de una extensa investigación desde principios del siglo XX. Debido a la complejidad del problema, hasta la década de los 60 las posibilidades de resolución eran muy escasas. Los métodos destinados a buscar programas óptimos se limitaban a problemas con un sólo centro de trabajo; o con simplificaciones muy rigurosas, como el caso "flow shop" permutacional con 3 máquinas como máximo. En el taller real, lo usual era ejecutar las actividades según determinadas reglas de prioridad. A partir de ese momento la situación cambio. Con el desarrollo de la microinformática, las posibilidades de tratamiento del problema aumentaron notablemente, pero desafortunadamente la complejidad computacional propia del problema seguía dificultando la resolución problemas prácticos en tiempos realistas. Ante este hecho, la investigación en este campo no ha dejado de producir nuevos procedimientos, pertenecientes un alto espectro de tipologías. En la actualidad, la búsqueda de técnicas distribuidas que puedan aprovechar varios recursos informáticos, ha llevado a proponer aproximaciones inspiradas en el funcionamiento de mercados competitivos. En este artículo se expone los fundamentos de estas técnicas y su relación con métodos de descomposición de programación matemática.

Palabras claves: *Programación Job Shop, subastas, relajación lagrangiana, descomposición de Dantzing-Wolfe, descomposición de Benders*

1. La programación "*Job Shop*"

La programación de operaciones consiste en la obtención de calendarios de producción, en los que se refleje las tareas a realizar, en qué momento y qué recursos se utilizarán para ello. Generalmente existen muchos programas posibles, pero no todos conducen al cumplimiento de objetivos (puede que no exista ninguno que lo haga). Por lo tanto, el establecimiento de los programas tendrá una marcada influencia en la eficiencia del sistema.

La dificultad de la programación de operaciones depende de la configuración de la planta de producción, pero en el caso más complejo (fabricación intermitente tipo "*Job Shop*"), el problema resulta de alta complejidad computacional (se le supone NP-Duro). En este caso el problema se denomina JSSP (*Job Shop Scheduling Problem*). Consiste en obtener el programa de producción que optimizando una determinada medida de eficiencia, destinado a fabricar N órdenes, que deben ser procesados en M centros de trabajo. Las órdenes se componen de una serie de operaciones, cada una de las cuales se debe procesar en un único centro de trabajo, establecido con antelación. La secuencia de realización de las operaciones es también única para cada orden y establecida sin ninguna ambigüedad con anterioridad a la programación. Es decir: las rutas de fabricación son únicas y conocidas de antemano. Se supone además, que existen almacenes locales de capacidad infinita en cada centro de trabajo, que los tiempos de preparación las operaciones se incluyen dentro de los tiempos de proceso, y que los tiempos de transporte son despreciables.

Ante la complejidad computacional que ha mostrado la programación "Job Shop", la investigación en este campo no ha dejado de producir métodos de resolución, pertenecientes un alto espectro de tipologías. Durante mucho tiempo estas se basaban fundamentalmente en métodos heurísticos y metaheurísticos, pero en la actualidad están apareciendo nuevas aproximaciones. Una de ellas es la basada en subastas.

2. Formulación Matemática del JSSP.

El JSSP se puede formular mediante modelos de optimización (función objetivo más restricciones), que variarán en función del sistema concreto que se esté programando y de los métodos de resolución que se desee utilizar. En esta sección se planteará un modelo matemático para un sistema de fabricación "*Job Shop*", mediante una formulación que por sus características nos será útil más adelante. Una aspecto importante de esta formulación es que en ella, el tiempo está discretizado en K instantes.

2.1. Descripción del Problema

Consideremos un sistema de fabricación compuesto por $\boldsymbol{H}$ centros de trabajo. La capacidad de cada centro de trabajo (**h**) en un instante de tiempo ($\boldsymbol{k}$) se denotará por $\boldsymbol{M_{hk}}$ para (h=1,2,$\cdots$,H) y (k=0,1,$\cdots$,K) siendo ($\boldsymbol{K}$) el máximo instante de tiempo considerado.

En el sistema de fabricación se deben fabricar $\boldsymbol{I}$ pedidos u órdenes. Cada orden ($\boldsymbol{i}$) tendrá asociada una fecha de llagada a la planta ($\boldsymbol{B_i}$), una fecha de entrega ($\boldsymbol{D_i}$) y un peso ($\boldsymbol{w_i}$) que denote la importancia de cumplir con esa fecha de entrega. La fabricación de una determinada orden ($\boldsymbol{i}$) conlleva la realización de $\boldsymbol{N_i}$ operaciones. Cada operación ($\boldsymbol{j}$) de cada orden ($\boldsymbol{i}$) se indica mediante los subíndices ($\boldsymbol{i,j}$) con (i=1,2,$\cdots$,I) y (j=1,2,$\cdots$,N_i). ($\boldsymbol{j}$) indica la posición de la operación dentro de la secuencia de operaciones necesaria para fabricar la orden (**i**). La máquina ($\boldsymbol{h}$) donde se realiza la operación (**j**) de la orden (**i**) se denotará por **h(i,j)**, y la duración de esa operación por $\boldsymbol{d_{ij}}$

2.1. Variables

El objetivo de la programación es determinar las fechas de comienzo y finalización de las operaciones que ha de realizar cada máquina. Dado que la formulación matemática del problema se corresponde con uno de optimización, donde las fechas son el resultado, estas deben corresponderse con las variables del problema. El instante de tiempo de comienzo de la operación (i,j) estará representado por la variable $\boldsymbol{b_{ij}}$, mientras que el instante de finalización se lo estará por $\boldsymbol{c_{ij}}$. Además se usará una variable auxiliar denominada $\boldsymbol{\delta_{ijkh}}$ cuyo valor será uno, si la operación (j) de la orden (i) se está ejecutando en el intervalo [k+1) en la máquina (h). Si no es así su valor será cero. Aunque estos conjuntos de variables son redundantes (con uno de los tres quedaría definido el programa de producción), su utilización facilitara la formulación del problema.

2.1. Función Objetivo

La función objetivo será una función matemática, que devolverá el valor de un determinado criterio de funcionamiento, en función de las variables que definen el programa de producción. Dependiendo de cual sea el criterio de funcionamiento que se desee mejorar se puede optar entre diferentes funciones objetivo. A titulo de ejemplo nosotros proponemos la siguiente (el razonamiento se puede extender a otro tipo de funciones objetivo):

$$\min \sum_{i=1}^{I} w_i \cdot T_i^2 \qquad con\ T_i = \max\left(0, (\max_j(c_{ij}) - D_i)\right) \tag{1}$$

2.1. Restricciones

Las dos restricciones fundamentales a las que esta sujeto el programa de producción se deben a la capacidad limitada de los recursos y al orden de las operaciones. Pero además, en esta formulación se introducirán otro tipo de restricciones, con objeto de ligar las variables δ_{ijkh}, c_{ij}, m_{ij} y b_{ij}.

- Restricciones de capacidad: la suma de todas las operaciones (i,j) que se realicen en el momento (k), en un centro de trabajo (h), deberá ser menor que la capacidad, en ese mismo instante (k), del centro de trabajo (h).

$$\sum_{i=1}^{I} \sum_{j=1}^{N_i} \delta_{ijkh} \le M_{kh} \qquad h = 1,2,\cdots,M \qquad k = 0,1,\cdots,K \tag{2}$$

- Restricciones en el orden de las operaciones: la operación (j+1) correspondiente a una orden (i) sólo podrán comenzar cuando haya finalizado la operación (j) de dicha orden.

$$c_{ij} + 1 \le b_{i,j+1} \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,(N_i - 1) \tag{3}$$

  Además se deberá cumplir que el comienzo de la primera operación no será anterior a la llegada de la orden al taller.

$$B_i \le b_{i,1} \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,(N_i - 1) \tag{4}$$

- Relaciones entre c_{ij} , b_{ij} y δ_{ijkh}. Dado que un programa de producción está completamente determinado por las fechas de comienzo cada operación (b_{ij}), será necesario ligar c_{ij} y δ_{ijkh} con b_{ij}.

$$c_{ij} = b_{ij} + d_{ij} \qquad i = 1,2,\cdots,I \quad j = 1,2,\cdots,N_i \tag{5}$$

$$\delta_{ijkh} = \begin{cases} 1 \quad si \quad k \in \lfloor b_{ij}, c_{ij} \rfloor \ \ y\ h = h(i,j) \\ 0 \quad en\ caso\ contrario \end{cases} \qquad \begin{array}{l} k = 0,1,\cdots,K \\ i = 1,2,\cdots,I \\ j = 1,2,\cdots,N_i \\ h = 1,2,\cdots,H \end{array} \tag{6}$$

3. Subastas para el JSSP.

Muchos problemas prácticos de optimización combinatoria, entre ellos el JSSP, pueden ser considerados como problemas combinatorios de asignación de recursos (CAP "*Combinatorial Allocation Problem*") [1]. En éstos, un conjunto de bienes (recursos) no divisibles, deben ser asignados a diferentes usuarios (agentes), de forma que la utilidad global del reparto sea máxima [1].

3.1 Problemas Combinatorios de Asignación de Recursos.

Formalmente el CAP se puede formular como sigue: sea (G) un conjunto de (n) bienes indivisibles y sea (A) un conjunto de (m) agentes que obtienen una determinada utilidad del uso de los bienes. Considérese además otro agente imaginario (Δ), que a partir de ahora llamaremos vendedor, poseedor inicial de estos bienes. Sea (F_a) un subconjunto del conjunto de bienes (G), sea $u_j(F_j)$ la utilidad que obtiene el agente (j) del uso de los bienes del subconjunto (F_j), y sea (q_i) el valor que el agente (Δ) da al bien (i). La solución de este problema consistirá en encontrar una partición[1] (f) de (G) $\{F_1,F_2,\cdots,F_m,F_\Delta\}$, tal que la suma de las utilidades de cada agente sea máxima. F_Δ representa los bienes no asignados a los agentes de (A), que quedan en propiedad del vendedor.

$$\begin{aligned}&\underset{F_i}{Max} \qquad \sum_{i\in F_\Delta} q_i + \sum_{j=1}^{m} u_j(F_j) \\ &Sujeto\ a \quad \{F_1, F_2, \cdots, F_m, F_\Delta\}\ sea\ \ partición\ de\ G\end{aligned} \tag{7}$$

El JSSP puede formularse como un CAP mediante la siguiente analogía: (1) los recursos se corresponden con intervalos de tiempo de uso de las máquinas (ejemplo: el intervalo [k, k+1) en la máquina (h)), (2) los agentes serán las órdenes de producción, que deben repartirse los intervalos de tiempo disponibles en las diferentes máquinas (recursos) para ser fabricadas, (3) la utilidad que cada orden de producción[2] (j) obtendrá por cada conjunto de recursos adquiridos (F_j), será función de sus restricciones, fechas de entrega, etc, (4) el valor (q_i) que el vendedor asigna a cada bien (i), puede ser considerado cero. Para la formulación del JSSP del apartado anterior una posible función de utilidad sería la de la expresión (8).

$$u_j(F_j) = \begin{cases} -w_j \cdot T_j^2(F_j) & si \quad F_j \quad no\ viola\ las\ restricciones\ (3)\ y\ (4) \\ -\infty & en\ caso\ contrario \end{cases} \tag{8}$$

$$con \qquad T_i(F_j) = \max\Big(0, (\max(k\ asignados\ a\ \ j) + 1 - D_j)\Big)$$

3.2 Precios de Equilibrio.

La ventaja de esta formulación es que posee cierta similitud con un mercado donde, (m) agentes intentan maximizar su utilidad adquiriendo bienes a un vendedor (Δ) que intenta maximizar su beneficio. Esta analogía va a permitir introducir en el problema otras variables, imprescindibles en todo mercado, que pueden ayudar a establecer nuevos procedimientos de resolución para los CAP. Dichas variables serán los precios de los bienes $\overline{p} = (p_1, p_2, \cdots, p_n)$. Con dichos precios se podrá definir el excedente que cada agente obtiene de una asignación (F_j) para unos precios dados como:

$$e_j(\overline{p}, F_j) = u_j(F_j) - \sum_{i\in F_j} p_i \qquad excedente\ de\ i \in\ A \tag{9}$$

---

[1] Se entiende por partición de un conjunto G, a un conjunto de subconjuntos $\{F_1,F_2,\cdots F_K\}$ de G, tal que ($F_1 \cup F_2 \cup F_3 \cup \cdots \cup F_k$) = G, ($F_i \cap F_j$) = conjunto vacío.

[2] En la formulación matemática del JSSP el subíndice utilizado para identificar a las órdenes era (i). En esta formulación del CAP, las órdenes que se corresponden con los agentes se identifican con (j).

$$e_\Delta(\overline{p}, F_\Delta) = \sum_{i \notin F_\Delta} (p_i - q_i) \qquad \textit{excedente de } \Delta \tag{10}$$

Si sumamos los excedentes obtenidos por todos los agentes, para un conjunto de precios ($\overline{p}$) y unas asignaciones $\{F_1, F_2, \cdots F_m, F_\Delta\}$, más la utilidad total del sistema antes de la asignación $(q_i+q_2+\cdots+q_n)$, obtendremos la utilidad total para la partición.

$$\begin{aligned}
&\sum_{j=1}^{m} e_j(\overline{p}, F_j) + e_\Delta(\overline{p}, F_\Delta) + \sum_{i=1}^{n} q_i = \sum_{j=1}^{m}\left(u_j(F_j) - \sum_{i \in F_j^*} p_i \right) + \sum_{i \notin F_\Delta^*} (p_i - q_i) + \sum_{i=1}^{n} q_i = \\
&= \sum_{j=1}^{m} u_j(F_j) - \sum_{j=1}^{m} \sum_{i \in F_j^*} p_i + \sum_{i \notin F_\Delta^*} p_i - \sum_{i \notin F_\Delta^*} q_i + \sum_{i=1}^{n} q_i = \sum_{j=1}^{m} u_j(F_j) + \sum_{i \notin F_\Delta^*} q_i
\end{aligned} \tag{11}$$

Sea $(F_s | \overline{p})$ la asignación de bienes al agente (s) ($s=1,2,\cdots,m,\Delta$), que maximiza el excedente de dicho agente para unos precios determinados ($\overline{p}$: $p_i \geq q_i$). Supongamos que existe un vector de precios ($\overline{p}^e$) para los cuales las asignaciones $\{F_1 | \overline{p}^e,\ F_2 | \overline{p}^e,\ \cdots F_m | \overline{p}^e,\ F_\Delta | \overline{p}^e\}$ constituyen una partición de (G). Estos precios se denominarán precios de equilibrio, y con ellos el resultado de (11) valido para cualquier partición de (G), se podrá rescribir como:

$$\sum_{j=1}^{m} u_j(F_j) + \sum_{i \notin F_\Delta^*} q_i = \sum_{j=1}^{m} e_j(\overline{p}^e, F_j) + e_\Delta(\overline{p}^e, F_\Delta) + \sum_{i=1}^{n} q_i \tag{12}$$

Dado que las asignaciones $\{F_1 | \overline{p}^e,\ F_2 | \overline{p}^e,\ \cdots F_m | \overline{p}^e,\ F_\Delta | \overline{p}^e\}$ maximizan cada uno de los sumandos situados a la izquierda de la igualdad, maximizarán también su total (función objetivo del CAP). Como además dichas asignaciones constituyen una partición de (G), serán por lo tanto la solución óptima del problema inicial.

Según este resultado, si se encontrara unos precios de equilibrio, el problema se podría resolver, maximizando de forma independiente los excedentes de los (m) agentes de (A). Esto es debido a que los precios de equilibrio poseen una información implícita de la escasez de cada bien y de su utilidad global. Cuando un agente maximiza su excedente, intentará no usar los bienes de precio elevado que no le aporten suficiente utilidad, quedándose este bien libre para otro agente obtenga más utilidad de él.

La posibilidad de obtener la asignación optima del CAP, maximizando de forma independiente los excedentes de cada agente, simplifica notablemente su resolución. Ante este hecho, la cuestión que inmediatamente surge es si existen los precios de equilibrio y en el caso de que existan ¿cómo calcularlos?. En cuanto a la existencia de los precios de equilibrio se pueden encontrar demostraciones que lo aseguran para funciones de utilidad $u(F_j)$ lineales. Pero también existen resultados que muestran, que cuando se da complementariedad[3] en la utilidad del uso de varios recursos, los precios de equilibrio pueden no existir. Este es el caso del JSSP: solo tiene utilidad para las órdenes la asignación de todos los recursos necesarios para su fabricación. La asignación de un solo recurso (sin el resto de los necesarios) no aporta ninguna utilidad a la orden

[3] Dos bienes son complementarios, si su uso conjunto aporta más utilidad, que la suma de la utilidad que aportan por separado.

3.2 Métodos de Resolución Basados en Subasta.

La analogía realizada entre los problemas de asignación combinatorios y los mercados, no solo facilita fragmentar el problema. Permite además, diseñar mecanismos competitivos de establecimiento de precios de equilibrio. En muchos mercados, la asignación de bienes y los precios a los que se realiza esta asignación, se fijan tras un proceso iterativo, basado en sucesivas propuestas y aceptaciones de precios por parte de compradores y vendedores. Dicho proceso se denomina subasta y su aplicación en los CAP es directa. Un ejemplo podría ser el siguiente: (1) el vendedor propone unos precios mínimos para todos los bienes, (2) teniendo en cuenta los precios, los agentes seleccionan los bienes que maximizan su excedente, (3) cuando un bien es seleccionado el vendedor incrementa su precio, (4) si se ha incrementado algún precio, volver al paso 2. El proceso se repite hasta que tras un incremento de precios ninguno de ellos es aceptado. Entonces los precios finales de los bienes serán los mayores de los aceptados hasta el momento.

Se puede demostrar que con el mecanismo citado en el párrafo anterior, es posible llegar a precios de equilibrio, en problemas con funciones de utilidad lineales. Pero éste no es el caso de muchos CAP, entre ellos el JSSP. Este hecho, ha convertido al diseño de subastas en un campo de estudio muy activo, que en la actualidad se concreta en dos líneas de investigación: una inspirada en los mercados (aproximación *bottom-up*) y otra derivada de la descomposición de problemas globales (aproximación *top-down*).

En las sociedades humanas se pueden encontrar mercados muy eficientes, capaces de establecer precios con relativa rapidez. Esta observación inspira la primera metodología de diseño de subastas. Mediante el estudio de mercado eficientes es posible extraer los mecanismos subyacentes (fijación de precios mínimos, secuencias de interacción, comportamientos individúeles, etc) de los que emerge el resultado final. Dichos mecanismos (simplificados o extendidos) pueden ser la base de modelos formales de subastas para el CAP. Establecido el modelo formal, habría que mostrar (mediante simulación) o demostrar (matemáticamente), que su dinámica conduce a precios de equilibrio y por lo tanto a la resolución del problema.

4. Subastas Derivadas de Métodos de Descomposición.

Una posibilidad de diseño de subastas, muy extendida en problemas complejos como el JSSP, es mediante descomposición del problema original en diferentes subproblemas, fácilmente identificables con agentes económicos maximizadores de beneficios. Entre los métodos de descomposición, cabe destacar: la *Relajación Lagrangiana* [2] y la descomposición de *Dantzing-Wolfe* [3]. Nosotros veremos sólo la primera (de la segunda se pueden obtener conclusiones muy parecidas).

4.1 Relajación Lagrangiana.

Las relajaciones son técnicas destinadas a simplificar las restricciones de los problemas de optimización. Se dice que un problema B {Min $f'(\bar{x})$ con $\bar{x} \in S'$} es relajación de un problema original A {Min $f(\bar{x})$ con $\bar{x} \in S$} cuando el conjunto de soluciones factibles de A esta contenido en el conjunto de soluciones factibles de B ($S \subset S'$) y además $f'(\bar{x}) \leq f(\bar{x})$. Si el problema relajado resulta más sencillo que el original, permitirá obtener de forma rápida cotas inferiores del óptimo. Esta posibilidad convierte a las relajaciones en técnicas de gran utilidad a la hora de evaluar la bondad de otro tipo de metodologías aproximadas de optimización.

Pero no solo tienen este aplicación. Las técnicas de relajación se pueden usar, para acelerar la convergencia de otros métodos, para diseñar algoritmos eficientes y para descomponer problemas en subproblemas más sencillos.

Un ejemplo de relajación es la “relajación lineal continua”, que elimina de los problemas de programación entera la imposición de integrabilidad de soluciones. No obstante hay otras relajaciones: “relajación por eliminación” (elimina restricciones), “relajación subrogada” (sustituye un conjunto de restricciones por una combinación lineal de éstas), “relajación lagrangiana”, etc. De todas ellas, la “relajación lagrangiana” es la que más interés está generando a la hora de abordar la programación “*Job Shop*”

Al igual que la relajación subrogada, la técnica de relajación lagrangiana propone sustituir un conjunto de restricciones por una combinación lineal de ellas. Sin embargo en esta última la combinación de restricciones no se considera como tal, sino que se introduce en la función objetivo, de forma que la violación de las restricciones penaliza el valor de dicha función. Apliquemos está técnica a la formulación del JSSP realizada en le apartado 2 y resumida a continuación (la restricción (14.d) engloba las de las expresiones (5) y (6) de la formulación inicial).

$$\min_{c_{ij} b_{ij} \delta_{ijkh}} \quad \sum_{i=1}^{I} w_i \cdot T_i^2 \tag{13.a}$$

$$\text{sujeto a} \quad \sum_{i=1}^{I} \sum_{j=1}^{N_i} \delta_{ijkh} \le M_{kh} \qquad h = 1,2,\cdots,M \qquad k = 0,1,\cdots,K \tag{13.b}$$

$$c_{ij} + 1 \le b_{i,j+1} \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,(N_i - 1) \tag{13.c}$$

$$B_i \le b_{i,1} \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,(N_i - 1) \tag{13.d}$$

$$\text{Re}\,laciones\ entre\ \text{var}\,iables \tag{13.e}$$

Relajemos las (M x K) restricciones de (14.b), multiplicándolas por un escalar ($\lambda_{hk} \ge 0$) e introduciéndolas en la función objetivo. Obtendremos el problema de la expresiones (14).

$$\min_{c_{ij} b_{ij} \delta_{ijkh}} \quad \sum_{i=1}^{I} w_i \cdot T_i^2 + \sum_{h=1}^{M} \sum_{k=0}^{K} \lambda_{hk} \left(\sum_{i=1}^{I} \sum_{j=1}^{N_i} \delta_{ijkh} - M_{kh} \right) \qquad \lambda_{hk} \ge 0 \tag{14.a}$$

$$\text{sujeto a} \quad c_{ij} + 1 \le b_{i,j+1} \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,(N_i - 1) \tag{14.b}$$

$$B_i \le b_{i,1} \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,(N_i - 1) \tag{14.c}$$

$$\text{Re}\,laciones\ entre\ \text{var}\,iables \tag{14.d}$$

Reagrupando los términos de la función objetivo el problema queda:

$$\min_{c_{ij} b_{ij} \delta_{ijkh}} \quad -\sum_{h=1}^{M} \sum_{k=0}^{K} \lambda_{hk} M_{kh} + \sum_{i=1}^{I} \left(w_i \cdot T_i^2 + \sum_{j=1}^{N_i} \sum_{h=1}^{M} \sum_{k=0}^{K} \lambda_{hk} \delta_{ijkh} \right) \qquad \lambda_{hk} \ge 0 \tag{15.a}$$

$$\text{sujeto a} \quad c_{ij} + 1 \le b_{i,j+1} \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,(N_i - 1) \tag{15.b}$$

$$B_i \le b_{i,1} \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,(N_i - 1) \tag{15.c}$$

$$\text{Re}\,laciones\ entre\ \text{var}\,iables \tag{15.d}$$

Formulando el problema dual de (15) se llega a la siguiente expresión, donde la variables de decisión pasan a ser los multiplicadores (λ_{hk}).

$$\max_{\lambda_{hk}} \quad L(\bar{\lambda}) = -\sum_{h=1}^{M}\sum_{k=0}^{K}\lambda_{hk} M_{kh} + \min_{c_{ij}b_{ij}\delta_{ijkh}}\left(\sum_{i=1}^{I}\left(w_i \cdot T_i^2 + \sum_{j=1}^{N_i}\sum_{h=1}^{M}\sum_{k=0}^{K} \lambda_{hk}\delta_{ijkh}\right)\right) \qquad (16.a)$$

$$\textit{sujeto a} \quad c_{ij} + 1 \le b_{i,j+1} \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,(N_i - 1) \qquad (16.b)$$

$$B_i \le b_{i,1} \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,(N_i - 1) \qquad (16.c)$$

$$\text{Re}\,\textit{laciones entre}\ \text{var}\,\textit{iables} \qquad (16.d)$$

$$\lambda_{hk} \ge 0 \qquad (16.c)$$

Para resolver este problema se propone el siguiente proceso iterativo: (1) seleccionar al azar los multiplicadores ($\lambda_{kh}{}^{0}$); (2) para un conjunto de multiplicadores dados[4] ($\lambda_{kh}{}^{n}$), que en la primera iteración son ($\lambda_{kh}{}^{0}$), obtener las variables (c_{ij}, b_{ij}, δ_{ijkh}) mediante la resolución del problema 17; (3) suponiendo que las variables de decisión toma los valores ($c^{n}{}_{ij}$, $b^{n}{}_{ij}$, $\delta^{n}{}_{ijkh}$), obtener nuevos valores para los multiplicadores ($\lambda_{kh}{}^{n+1}$); (4) con los nuevos multiplicadores actualizados volver al paso (2).

$$\min_{c_{ij}b_{ij}\delta_{ijkh}} \quad \sum_{i=1}^{I}\left(w_i \cdot T_i^2 + \sum_{j=1}^{N_i}\sum_{h=1}^{M}\sum_{k=0}^{K} \lambda_{hk}\delta_{ijkh}\right) \qquad (17.a)$$

$$\textit{sujeto a} \quad c_{ij} + 1 \le b_{i,j+1} \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,(N_i - 1) \qquad (17.b)$$

$$B_i \le b_{i,1} \qquad i = 1,2,\cdots,I \qquad j = 1,2,\cdots,(N_i - 1) \qquad (17.c)$$

$$\text{Re}\,\textit{laciones entre}\ \text{var}\,\textit{iables} \qquad (17.d)$$

Por la propia estructura del problema (17), éste se puede descomponer fácilmente en (I) subproblemas (expresión 18), lo que se simplifica notablemente la obtención de ($c^{n}{}_{ij}$, $b^{n}{}_{ij}$, $\delta^{n}{}_{ijkh}$) dados los multiplicadores ($\lambda_{k}{}^{n}$). Obsérvese como cada subproblema se puede identificar con cada orden (i).

$$\min_{c_{ij}b_{ij}\delta_{ijkh}} \quad w_i \cdot T_i^2 + \sum_{j=1}^{N_i}\sum_{h=1}^{M}\sum_{k=0}^{K}\lambda_{hk}^{n}\delta_{ijkh} \qquad (18.a)$$

$$\textit{sujeto a} \quad c_{ij} + 1 \le b_{i,j+1} \qquad j = 1,2,\cdots,(N_i - 1) \qquad (18.b)$$

$$B_i \le b_{i,1} \qquad j = 1,2,\cdots,(N_i - 1) \qquad (18.c)$$

$$\text{Re}\,\textit{laciones entre}\ \text{var}\,\textit{iables} \qquad (18.d)$$

Para calcular los multiplicadores en cada iteración, se pueden utilizar varias técnicas. Una de ellas de las más usadas consiste en aplicar el método del subgradiente para maximizar la función objetivo de (16.a). Así los multiplicadores se pueden actualizar según la siguiente expresión:

[4] El superíndice (n) hace referencia a la iteración

$$\lambda_{hk}^{\;n+1} = \lambda_{hk}^{\;n} + \alpha^n \cdot \qquad con \quad \bar{d}^{\,n} : vector\ gradiente\ de\ L(\bar{\lambda}) \tag{19}$$

$$\alpha^n : paso$$

$$d_{hk}^n = \frac{\partial\left(L^n(\bar{\lambda})\right)}{\partial \lambda_{hk}} = \sum_{i=1}^{I}\sum_{j=1}^{N_i} \delta_{ijkh} - M_{hk}$$

Si se analiza podrá observar que todos se asemeja a un procesos de subasta, donde cada subproblema puede ser considerado un agente maximizador de beneficios (minimizado de costes). Los multiplicadores λ_{hk} se podrían interpretar como el precio de el intervalo de tiempo [k,k+1) en la máquina h. Esta analogía, unida a la convergencia demostrada de los procedimientos por descomposición, permite el diseño de subastas eficientes para los CAP.

El proceso de subasta derivado de esta técnica quedaría como sigue: (0) se proponen precios ($\lambda_{kh}{}^{0}$) para todos los intervalos de tiempo de uso de las máquinas, (1) con los precios propuestos se calcula para cada orden (i) la combinación de recursos (δ_{ijkh}) que minimice la expresión 18 (costes de recursos más penalización por retraso), (2) las soluciones obtenidas de todos los subproblemas en el paso 1, pueden no constituir una solución global factible (un intervalo de tiempo [k,k+1) en una máquina (h) puede tener asignados más trabajos que los que puede realizar M_{hk}), (3) se recalculan los precios ($\lambda_{kh}{}^{n}$) utilizando para ello la expresión 19 (obsérvese como los precios se aumentan cuando las asignaciones parciales realizadas en el paso 1 superan la disponibilidad M_{hk}), (4) si la solución encontrada hasta el momento no es de suficiente calidad entonces, volver al paso 1.

5 Conclusiones.

Una de las propuestas más novedosas de resolución del JSSP se basa en una aproximación inspirada en mercados competitivos. Formulando el JSSP como un problema combinatorio de asignación de recursos, se pueden llegar a mecanismos de resolución basados en subastas. En el problema así formulado, los intervalos de tiempo de uso de máquinas se corresponderían con los recursos y las órdenes de producción con los agentes a los que se debe asignar estos recursos. Mediante mecanismos de subastas, en los que de forma iterativa se van proponiendo, asignaciones de recursos y precios de estos, se puede llegar a soluciones globalmente optimas. Aunque de partida, estos métodos parezcan diferentes a los métodos de descomposición, en realidad son muy parecidos ya que estos últimos pueden ser interpretados en términos de subastas.

Al igual que los métodos de descomposición, los basados en subasta tiene la ventaja de permitir la distribución del cálculo, en procesadores separados (entidades que intervienen en la subasta: órdenes y máquinas). Cada unidad de cálculo, no necesitará de todos los datos del problema original. Le bastará con cierta información propia (a las órdenes su función de utilidad y a las máquinas su disponibilidad) más los precios (que contendrán de alguna forma la información global que sea de interés). La posibilidad de distribución del cálculo y de los datos de partida, no solo tiene interés desde el punto de vista de la potencia de cálculo (se gana capacidad de cálculo al poder utilizar varias computadoras simultáneamente), es interesante también desde el enfoque de la arquitectura, desde el punto de vista del desarrollo del sistema software, y desde la problemática de la ejecución y control de sistemas de fabricación.

Referencias

[1] Welman, Michael P.; Walsh, William E.; Wurman, Peter R.; MacKie-Mason, Jeffrey K. (2001), "Auction Protocols for Decentralized Scheduling", *Games and Economic Behaviour,* 35, pp 271-303

[2] Christos A. Kaskavelis and Michael C. Caramanis (1998). "Eficient Lagrangian relaxation algorithms for industry size job-shop scheduling problems", *IIE Transactions* (1998) 30, pp 1085-1097.

[3] Davies, Ryan W (2005),. "Distributed Generalized Vickrey Auctions Based on the Dantzig-Wolfe and Benders Decomposition Methods for Linear Programs", *Thesis, Bachelor of Arts, Harvard College, Cambridge, Massachusetts*