Qüestiió - V. 8, n.º 4 (desembre 1984) pp. 197-205

COTAS INFERIORES PARA EL PROBLEMA DE SECUENCIACIÓN CON RESTRICCIONES SOBRE LOS RECURSOS

R. ÁLVAREZ - VALDÉS, J. M. TAMARIT
UNIVERSIDAD DE VALÉNCIA

El trabajo explora dos vías de obtención de cotas inferiores para el problema de secuenciación de actividades con restricciones sobre los recursos, a partir de una formulación entera del problema. Una primera cota se obtiene de la relajación lineal y la aplicación sucesiva de planos de corte. El segundo método utiliza la relajación lagrangiana. El problema relajado se descompone en dos subproblemas para los que se proponen algoritmos de resolución. Se incluyen resultados computacionales que ilustran el comportamiento de las cotas obtenidas en ambos casos sobre una colección de problemas test.

Keywords: SCHEDULING, INTEGER PROGRAMMING, CUTTING PLANES, LAGRANGEAN RELAXATION.

1. INTRODUCCION.

El problema de secuenciación de un conjunto de operaciones $\{1,2,\ldots,n\}$ con limitación de recursos consiste en la determinación de un conjunto de tiempos $\{t_1,t_2,\ldots,t_n\}$ de inicio de dichas operaciones de forma que se satisfagan las posibles relaciones de precedencia entre ellas y las restricciones sobre los recursos disponibles para su ejecución.

Este problema puede formularse de la forma siguiente:

$$\text{Min } t_n \tag{1}$$

$$\text{sujeto a } \quad t_i - t_j \geqq d_{ij} \qquad \forall (i,j) \in H \tag{2}$$

$$\sum_{S(t)} r_{ik} \leqq b_k \qquad \begin{matrix} t=1,2,..,T \\ k=1,2,..,K \end{matrix} \tag{3}$$

donde: H es el conjunto de relaciones de precedencia.

d_{ij} el tiempo mínimo que ha de separar el inicio de i y j.

S(t) el conjunto de operaciones en proceso en tiempo t.

r_{ik} la cantidad de recurso k requerida por la operación i.

b_k la cantidad total del recurso k disponible.

El problema definido por (1) y (2) es el problema simple de secuenciación de actividades en un proyecto, resuelto eficientemente, incluso con miles de actividades, por las técnicas PERT y CPM. Las restricciones (3) introducen en el modelo la limitación de recursos precisos para realizar las operaciones. Con ellas el problema aumenta extraordinariamente en complejidad y requiere técnicas más sofisticadas de resolución. Antes de entrar en el contenido de nuestro trabajo daremos un breve resumen de los diferentes métodos propuestos en los últimos años, comentando sus éxitos y limitaciones.

Pritskers, Waters y Wolfe /12/ en 1969 proponen una formulación entera con tres tipos de variables 0-1, suficientemente flexible para admitir distintos tipos de función objetivo y restricciones especiales. Sin embargo, el número de variables crece muy rápidamente con el tamaño del problema y su algoritmo de programación lineal entera sólo puede ser usado en problemas muy pequeños.

- Ramón Alvarez-Valdés Olaguibel i José Manuel Tamarit Goerlich - Departament d'Estadistica i Investigació Operativa - Facultat de Ciencies Matematiques de la Universitat de Valencia.

- Article rebut el maig de 1984.

Davis y Heidorn /6/ proponen un algoritmo de branch & bound basado en un procedimiento de resolución del problema de la línea de montaje ideado por Gutjahr y Nemhauser /10/. Su algoritmo produce una familia de "conjuntos posibles", conjuntos de operaciones que pueden haber sido completadas en un tiempo t. El número de tales conjuntos crece muy rápidamente y el algoritmo sólo puede aplicarse a problemas con un número reducido de operaciones.

Patterson y Roth /11/ presentan una formulación entera en la que no tratan de reducir el número de variables, sino de dotar al problema de una estructura que permita un empleo eficiente de un algoritmo de enumeración implícita. Así se logra una reducción en las necesidades de memoria y en los tiempos de resolución.

En 1978, Talbot y Patterson /15/ usan la formulación anterior introduciendo la idea de los "network cuts", mecanismos que identifican en las etapas iniciales del proceso de enumeración aquellas secuencias parciales que no pueden conducir a una solución óptima. Con ello los tiempos de resolución se reducen significativamente.

También en 1978, Stinson, Davis y Khumawala /14/ crean un algoritmo de branch & bound con cotas basadas en las relaciones de precedencia y en las restricciones de recursos, y un conjunto muy elaborado de reglas de ramificación que permiten grandes reducciones en el árbol de búsqueda y buenos tiempos de resolución.

El único intento de aplicar las técnicas de la relajación lagrangiana a este problema se debe a Fisher /7/, /8/, en 1976. En su formulación las operaciones se agrupan en proyectos sin otra relación entre sí que el uso de los mismos recursos. Al relajar las restricciones sobre los recursos, utilizando multiplicadores de Lagrange, los problemas resultantes son sencillos de resolver, pero no lo es el de determinar multiplicadores que produzcan buenas cotas. Fischer propone un método basado en la generación de columnas que produce buenos resultados, pero no presenta una experiencia computacional completa que permita una comparación con los trabajos anteriormente citados.

Un enfoque completamente diferente es el propuesto por Balas/3/, basado en el concepto de "grafo disjunto", para el problema del "job shop". En este problema hay una sola máquina de cada tipo, por lo que dos operaciones i,j que necesiten la misma máquina no pueden procesarse a la vez. Para evitarlo, se añade un par disjunto $\{(i,j),(j,i)\}$ al grafo H de las relaciones de precedencia. Si se toma una selección S, consistente en exactamente un arco de cada par, el camino más largo en el grafo $S \cup H$ representa la duración de una solución posible para el problema original. Se trata, por tanto, de hallar la selección S que produzca el mínimo camino crítico. El mismo Balas /4/ generaliza esta idea para el problema que nos ocupa. Como b_k no es siempre igual a la unidad, dos operaciones que necesitan del recurso k no son necesariamente incompatibles. Por tanto, una selección S, para que sea posible, no necesita incluir un arco de cada par disjunto. El número de tales selecciones aumenta y la determinación de si una selección es posible no es sencilla. Balas propone ciertas condiciones que lo aseguran, pero no construye un algoritmo que resuelva el problema.

Gorenstein /9/ diseña un algoritmo que sigue las ideas de Balas. Para garantizar que una selección es posible propone un método basado en el cálculo del flujo máximo en un grafo bipartido. Este método no es eficiente, como señalan Bennington y McGinnis /5/, pero la cota alternativa que ellos proponen no está plasmada en un algoritmo de resolución del problema.

Teniendo en cuenta todos estos antecedentes, el trabajo que presentamos está dedicado a la obtención de cotas que puedan ser eficientes en una estructura de branch & bound, siguiendo dos líneas alternativas. En primer lugar, se introduce una formulación entera del problema, para usar como cota su relajación lineal. Esta primera cota es reforzada por la aplicación sucesiva de planos de corte que aproximan las soluciones del problema lineal resultante en cada caso a la solución entera. En la segunda parte del trabajo la misma formulación entera, ligeramente modificada, sirve como base para la relajación lagrangiana, que descompone el problema original en dos subproblemas más sencillos, para los que se proponen algoritmos de resolución. En ambos casos se incluyen resultados computacionales que muestran las cotas obtenidas sobre una colección de problemas test.

2. UNA FORMULACION ENTERA DEL PROBLEMA.

El problema puede formularse como un problema de programación lineal entera con variables 0-1 de la forma siguiente:

$$\text{Min} \quad \sum_t t y_{nt} \tag{4}$$

$$\text{sujeto a} \quad \sum_t y_{it} = 1 \qquad i=1,2,\ldots,n \tag{5}$$

$$\sum_t t(y_{jt}-y_{it}) \geq d_i \qquad \forall (i,j)\in H \tag{6}$$

$$\sum_t x_{it} = d_i \qquad i=1,2,\ldots,n-1. \tag{7}$$

$$\sum_i r_{ik} x_{it} \leq b_k \qquad \begin{array}{l} t=1,2,\ldots,T \\ k=1,2,\ldots,K \end{array} \tag{8}$$

$$\sum_{s=t-d_i+1}^{t} x_{is} \geq d_i y_{it} \qquad \begin{array}{l} i=1,2,\ldots,n-1. \\ t=1,2,\ldots,T. \end{array} \tag{9}$$

$$y_{it}, x_{it} \in \{0,1\} \tag{10}$$

donde

$$y_{it} = \begin{cases} 1 & \text{si la operación } i \text{ empieza en el tiempo } t \\ 0 & \text{en caso contrario} \end{cases}$$

$$x_{it} = \begin{cases} 1 & \text{si la operación } i \text{ está en proceso en tiempo } t \\ 0 & \text{en caso contrario} \end{cases}$$

d_i = duración de la operación i.
H = conjunto de relaciones de precedencia.
r_{ik} = unidades del recurso k requeridas por la operación i.
b_k = disponibilidad total del recurso k.
T = cota superior para la duración total del proyecto.

Las restricciones 5 y 7 aseguran respectivamente el comienzo y la realización completa de cada operación. Las restricciones 6 reflejan las relaciones de precedencia entre operaciones. Las restricciones 8 garantizan en todo tiempo que los recursos requeridos por las operaciones en proceso en ese tiempo no sobrepasan el total de recursos disponibles. Por último, (9) son restricciones de consistencia que aseguran que en el tiempo en el que una operación comienza dicha operación está en proceso y lo estará ininterrumpidamente hasta su terminación.

La función objetivo es el tiempo de iniciación de la operación n, operación ficticia sucesora de todas las operaciones del proyecto. En el tiempo en el que n puede comenzar, todas las restantes operaciones han sido totalmente procesadas.

La formulación expuesta utiliza (n-1)T variables x_{it} y nT variables y_{it}. Sin embargo, el número de variables puede reducirse mediante un uso adecuado de las relaciones de precedencia. Sea v(i,j) la longitud del camino más largo desde i hasta j y escribimos $r_i=v(1,i)$, $q_i=T-v(m,i)$, $q_i'=T-v(i,n)-d_i$. De esta forma, r_i es el primer tiempo en el que puede empezar la operación i, q_i el último tiempo en el que puede empezar i y q_i' el último tiempo en el que la operación i puede estar en proceso. Tendremos por tanto

$$y_{it} = 0 \quad \text{si} \quad t < r_i \text{ ó } t > q_i$$

$$x_{it} = 0 \quad \text{si} \quad t < r_i \text{ ó } t > q_i'$$

De esta forma reducimos sustancialmente el número de variables antes de comenzar la resolución del problema propiamente dicha. En los problemas test a los que se referían los resultados computacionales esta reducción osciló entre el 33 y el 80%, con una reducción media del 55%.

Como el objeto de nuestro trabajo es la obtención de cotas inferiores para su utilización en un algoritmo de branch & bound, vamos a explorar dos relajaciones de esta formulación entera. En el apartado 3 estudiaremos la relajación lineal y planos de corte, mientras que el apartado 4 está dedicado al estudio de la relajación lagrangiana.

3. RELAJACION LINEAL Y PLANOS DE CORTE.

En la formulación entera del apartado anterior relajamos la condición de integridad (10), obteniendo así un problema lineal cuya solución proporciona una cota inferior para el problema entero. Los resultados obtenidos sobre una colección de 20 problemas test a los que iremos haciendo referencia a lo largo de este trabajo, aparecen en la columna 2 de la Tabla 1. Si los comparamos con las columnas 1 y 9, que reflejan el valor del PERT y de la solución óptima de cada problema respectivamente, podemos comprobar que la cota lineal está muy lejos de la solución óptima entera y muy próxima a la del PERT, que se puede hallar de una forma computacionalmente más eficiente. Como se verá con más detalle en los siguientes apartados, las restricciones sobre los recursos (8), si bien suficientes para el problema entero, son fácilmente satisfechas por un conjunto de variables x_{it} no enteras que no producen aumentos significativos en la función objetivo.

Esto nos indujo a pensar en la introducción de planos de corte que, redundantes en la formulación entera, fueran relevantes en la relajación lineal y potenciaran la cota resultante. A continuación se detallan los diferentes tipos de planos de corte que incorporamos sucesivamente al problema lineal.

3.1. INCORPORACION DE LAS VARIABLES y_{nt} EN LAS RESTRICCIONES DE RECURSOS.

La función objetivo está en función únicamente de y_{nt}. Por ello el problema lineal trata de realizar la operación n, o parte de ella, lo antes posible, sin que las restricciones de precedencia, que serían suficientes en el problema entero, puedan impedirlo. De esta forma encontramos que en la solución lineal la operación n empieza antes que otras operaciones hayan acabado. Para evitar, en cierta medida, esta situación podemos considerar que esta operación n es incompatible con todas las demás, lo que escrito en términos de recursos consiste en añadir a (8) el término $y_{nt}b_k$, resultando

$$\sum_{i=1}^{n-1} r_{ik}x_{it} + \sum_{s=1}^{t} b_k y_{ns} \leq b_k \qquad \begin{array}{l} k=1,2,..,K \\ t=1,2,..,T \end{array} \qquad (11)$$

La influencia de esta restricción aparece en la columna 3 de la Tabla 1.

3.2. RESTRICCIONES DE CONSISTENCIA DE LAS x_{it}

Las restricciones sobre los recursos (8) se cumplen fácilmente en el problema lineal procesando una pequeña parte de cada operación en cada tiempo, es decir, dispersando la duración d_i en gran número de tiempos. Para contrarrestar esta dispersión, introducimos una relación de consistencia para las variables x_{it}. Basándonos en el hecho de que una operación i ha de realizarse en d_i tiempos consecutivos, podemos exigir que las x_{it} cumplan

$$\sum_{j=m,m+d_i,m+2d_i,..} x_{ij} = 1 \qquad \begin{array}{l} i=1,2,..,n-1 \\ m=1,2,..,d_i-1 \end{array} \qquad (12)$$

El resultado de incorporar estas restricciones aparece en la columna 4 de la Tabla 1

3.3. RESTRICCIONES DE CONSISTENCIA ENTRE x_{it} e y_{it}

Las restricciones 9 que relacionan x_{it} e y_{it} son el único nexo de unión entre las relaciones de precedencia y de recursos que configuran el problema. Lo que estas restricciones persiguen es que, una vez la operación i comienza, momento en el que $y_{it}=1$, la operación esté en proceso de forma ininterrumpida durante d_i tiempos, tal como refleja la figura 1.

				t	t+1		$t+d_i-1$		
y_{it}	0	0	0	1	0	. . . . .	0	0	. .
x_{it}	0	0	0	1	1	. . . . .	1	0	. .

Figura 1. Valores de y_{it} y x_{it} en la realización de i.

Podemos considerar estos tiempos en los que $x_{it}=1$ como los elementos de una progresión aritmética de razón unidad. La expresión de su suma, poniendo los miembros de la progresión en función de x_{it} y su primer y último elemento en función de y_{it}, nos da la restricción 13.

Igualmente podemos utilizar estos tiempos en los que $x_{it}=1$ como exponentes de 2 de forma que se obtenga una progresión geométrica de razón 2. De forma análoga a la ya descrita la suma de esta progresión da lugar a la restricción 14.

$$\sum_t t x_{it} = 1/2 \{\sum_t t y_{it} + (\sum_t t y_{it} + d_i - 1)\} d_i; \quad i=1,2,..,n-1 \quad (13)$$

$$\sum_t 2^t x_{it} = \sum_t 2^{t+d_i} y_{it} - \sum_t 2^t y_{it} \qquad i=1,2,..,n-1 \quad (14)$$

El efecto de estas nuevas restricciones aparece en la columna 5 de la Tabla 1.

3.4. CONJUNTOS DE OPERACIONES INCOMPATIBLES POR RECURSOS.

La forma de las restricciones sobre los recursos 8 es la más natural en la formulación entera del problema, pero es muy débil para el problema lineal. Por ejemplo, si $b_k=10$, $r_{ik}=5$, $r_{jk}=6$, las operaciones i,j son incompatibles. Sin embargo es posible tener $x_{it}=0.9$, $x_{jt}=0.9$ en el mismo t, obteniendo así valores de la función objetivo del problema lineal muy alejadas del valor óptimo de la solución entera. Más restrictivo para el problema lineal sería exigir que si $r_{ik}+r_{jk}>b_k$, entonces $x_{it}+x_{jt}\leq 1\ \forall t$.

Esta idea de pares de operaciones incompatibles por recursos puede extenderse a conjuntos de operaciones A tales que $\forall (i,j)\in A$ $r_{ik}+r_{jk}>b_k$ para algún k. Para cada conjunto A tendremos un conjunto de restricciones de la forma

$$\sum_{i\in A} x_{it} \leq 1 \qquad t=1,2,..,T. \qquad (15)$$

Aun antes de incorporar estas restricciones, $L = \max\{\sum_{i\in A} d_i\}$ es una cota inferior para la solución del problema, que puede ser significativa en problemas en los que las limitaciones de recursos sean muy restrictivas.

La influencia de las restricciones 15 aparecen en la columna 6 de la Tabla 1.

3.5. NUEVAS RELACIONES DE PRECEDENCIA.

De forma similar a lo comentado en 3.4, las restricciones de precedencia, tal como aparecen en (6), se satisfacen en el problema lineal mediante la dispersión, en este caso de las y_{it}. Para reforzar las relaciones de precedencia introducimos $\forall (i,j)\in H$.

$$\sum_{s=t}^{T} y_{is} + \sum_{r=1}^{t+d_i-1} y_{jr} \leq 1 \qquad t=1,2,..,T \qquad (16)$$

La incorporación de esta restricción aparece reflejada en la columna 7 de la Tabla 1.

3.6. RESULTADOS COMPUTACIONALES

La relajación lineal y los sucesivos planos de corte han sido probados sobre un conjunto de 20 problemas test, con un número de operaciones que oscila entre 12 y 20 y con 3 recursos para cada operación.

Los resultados obtenidos aparecen en la Tabla 1, cuyas columnas son:

(1) valor de la solución del PERT

(2) valor de la relajación lineal

(3)-(7) soluciones obtenidas al añadir acumulativamente los planos descritos en los apartados 3.1 a 3.5

(8) solución obtenida al incluir en las restricciones 15 del apartado 3.4 la incompatibilidad de operaciones debida a las relaciones de precedencia, junto a las incompatibilidades por recursos.

(9) solución óptima del problema entero.

De la Tabla 1 se desprende que la distancia relativa de la mejor cota obtenida a la solución óptima es en promedio de un 6.6%. Por tanto, las cotas parecen ser suficientemente potentes para producir reducciones significativas en el árbol de búsqueda del branch & bound.

La calidad de las cotas obtenidas al añadir todos los planos de corte (columna 8 de la Tabla 1) no se ve afectada por la estructura de las relaciones de precedencia y tampoco parece provocar variaciones la fuerza de las restricciones sobre los recursos, es decir, la mayor o menor escasez relativa de los mismos. Los resultados son similares en problemas poco restringidos, como los de las filas 1,2,3,4 ó 20, y en los fuertemente restringidos, como los de las filas 7,12, 13 ó 18. En este sentido puede hablarse de una cota estable en relación a los dos elementos fundamentales en la definición del problema.

4. COTAS INFERIORES DERIVADAS DE LA RELAJACION LAGRANGIANA.

El problema formulado en el apartado 2 puede reformularse adoptando como función objetivo

$$\text{Min} \sum_t t y_{nt} + \sum_t \sum_i 10 t x_{it} \qquad (17)$$

TABLA 1

Resultados de la relajación lineal y planos de corte

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
24	24.0	24.0	24.17	24.33	25.69	25.94	26.38	28
20	20.27	20.27	20.32	20.53	20.53	20.57	20.59	22
26	26.0	26.0	26.0	26.02	26.83	27.04	28.91	30
18	18.0	18.0	18.0	18.0	18.01	18.01	18.06	21
24	24.0	24.16	24.16	24.37	25.33	25.53	26.18	29
18	18.0	19.0	19.0	19.0	19.48	19.66	21.79	24
9	9.22	11.67	11.67	11.67	12.17	12.17	12.30	16
15	15.04	15.05	15.09	15.60	16.80	17.03	19.81	22
15	15.05	16.50	16.50	16.50	20.22	20.54	22.50	23
13	13.09	14.50	14.50	14.76	17.57	18.08	18.30	20
13	13.28	18.33	18.33	18.33	19.67	19.67	19.67	22
9	9.26	11.67	11.67	11.67	14.76	14.81	16.36	17
9	9.45	14.33	14.33	14.33	16.83	16.83	16.83	17
9	9.27	12.0	12.0	12.0	12.82	13.01	13.62	14
22	22.0	24.17	24.17	24.17	24.42	24.59	26.83	29
19	19.0	19.29	19.29	19.29	19.33	19.72	21.41	23
9	9.0	15.67	15.67	15.67	15.75	15.78	16.20	19
10	10.03	12.92	12.92	12.92	13.84	14.04	15.76	18
9	9.65	13.33	13.33	13.33	16.72	16.87	17.37	19
20	20.0	20.67	20.67	20.67	21.21	21.21	21.58	24

TABLA 2

Resultados computacionales de la Relajación Lagrangiana

(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
1	6924	7188	7545	11	2083	2582	2860
2	4710	4941	5071	12	759	1003	1071
3	5386	6035	6199	13	1009	1257	1381
4	3918	4017	4389	14	1039	1539	1664
5	6984	7576	8058	15	6622	6793	7103
6	2685	3009	3119	16	3939	4540	4810
7	1209	1396	1573	17	1099	1420	1441
8	1805	1963	1968	18	1039	1202	1552
9	1615	1747	1777	19	1340	1580	1764
10	1393	2097	2548	20	4390	4810	4981

y manteniendo como restricciones las ya descritas (5)-(10). Podemos utilizar esta formulación en lugar de la original puesto que cualquier solución óptima para este problema es óptima para el problema original. La razón para el cambio será comentada en 4.2.

Una forma natural de relajar lagrangianamente este problema es relajar las restricciones 9 que son las únicas que incluyen a la vez variables x_{it} e y_{it}. Si definimos un multiplicador λ_{it} para cada restricción obtenemos el problema

$$(PR_\lambda) \quad \text{Min} \sum_t ty_{nt} + \sum_t \sum_i 10tx_{it} + \sum_t \sum_i \lambda_{it}(y_{it}d_i - \sum_{s=t-d_i+1}^{t} x_{is})$$

sujeto a

$$\sum_t y_{it} = 1 \qquad i=1,2,..,n.$$

$$\sum_t t(y_{jt}-y_{it}) \geq d_i \qquad \forall(i,j)\in H$$

$$\sum_t x_{it} = d_i \qquad i=1,2,..,n-1$$

$$\sum_i r_{ik}x_{it} \leq b_k \qquad t=1,2,..,T \quad k=1,2,..,K$$

$$x_{it}, y_{it} \in \{0,1\} \qquad i=1,2,..,n \quad t=1,2,..,T$$

que puede descomponerse en dos problemas independientes, uno en variables x_{it} y otro en variables y_{it}. Así, definiendo

$$\alpha_{it} = \begin{cases} \lambda_{it}d_i & i\neq n \\ t & i=n \end{cases} \qquad t=1,2,..,T$$

$$\beta_{it} = 10t - \sum_{s=t}^{t+d_i-1} \lambda_{is} \qquad i=1,2,..,n-1 \quad t=1,2,..,T$$

minimizar (PR_λ) es equivalente a la minimización de (PR_α) y (PR_β), donde

$$(PR_\alpha) \quad \text{Min} \sum_{i,t} \alpha_{it}y_{it}$$

sujeto a

$$\sum_t y_{it} = 1$$

$$\sum_t t(y_{jt}-y_{it}) \geq d_i$$

$$y_{it}\in\{0,1\}$$

$$(PR_\beta) \quad \text{Min} \sum_{i,t} \beta_{it}x_{it}$$

sujeto a

$$\sum_t x_{it} = d_i$$

$$\sum_i r_{ik}x_{it} \leq b_k$$

$$x_{it}\in\{0,1\}$$

4.1. UN ALGORITMO DE BRANCH & BOUND PARA $(PR\alpha)$

El problema (PR_α) puede ser considerado una generalización del PERT en el que, junto a la minimización del tiempo total de ejecución del proyecto, se han de minimizar otros costes α_{it} relativos a la realización de la operación i en el tiempo t.

El algoritmo que proponemos para su resolución parte de una cota superior T para el tiempo total de realización del proyecto y resuelve k+1 subproblemas, con k = T - v(1,n), es decir, uno para cada valor de la holgura respecto al tiempo obtenido por el PERT. En el caso k=0 la solución es inmediata, pues el valor óptimo será $v(1,n) + \sum_i \alpha^*_{it}$, con

$$\alpha^*_{it} = \min_{r_i\leq t\leq q_i} \alpha_{it}$$

Para $k\neq 0$ el problema es:

$$\text{Min } v(1,n) + k + \sum_i \alpha_{it}y_{it}$$

sujeto a

$$\sum_t y_{it} = 1$$

$$\sum_t t(y_{jt}-y_{it}) \geq d_i$$

es decir, se trata de elegir para cada operación i un α^*_{it}, con $r_i\leq t\leq q_i+k$, de forma que se satisfagan las relaciones de precedencia y se minimice la suma de α^*_{it}.

Para ello hemos diseñado un algoritmo de branch & bound en el que en cada nudo se eligen las mínimas α_{it} para las operaciones i no fijadas, obteniendo una cota inferior. Si la solución así construida es posible o su valor supera el de la mejor solución conocida se realiza el backtrack. En caso contrario, la cota puede aumentarse mediante un procedimiento de reasignación de operaciones, de forma que se mantenga el carácter de cota inferior. Si la nueva solución tampoco es posible ni satura el nudo, se continúa el proceso de ramificación fijando una nueva operación a un tiempo de realización.

Este método consigue rápidamente soluciones posibles y produce árboles de búsqueda muy reducidos para cada subproblema. Asimismo ocurre con frecuencia que algunos subproblemas no necesitan ser totalmente resueltos, pues la mejor solución obtenida en la resolución de los subproblemas anteriores es menor que una cota inferior para aquéllos. La conjunción de estos factores produce un algoritmo sencillo y computacionalmente eficiente - para (PR_α).

Una explicación detallada del mismo, sus resultados computacionales y un método alternativo de solución basado en una segunda relajación pueden hallarse en Alvarez-Valdés y Tamarit /1/.

4.2. UN ALGORITMO DE BRANCH & BOUND PARA $(PR\beta)$

El problema (PR_β) puede ser interpretado como una generalización del problema de asignación generalizado (GAP), tal como aparece en Ross y Soland /13/. La diferencia de (PR_β) respecto al GAP reside en la existencia de tareas múltiples (aquí $\sum_t x_{it}$ es d_i y no 1), y la posibilidad de incorporar más de una restricción sobre los recursos.

El algoritmo utilizado está basado en el algoritmo de branch & bound propuesto por Ross y Soland /13/ para el GAP, incorporando las modificaciones derivadas de las especiales características de nuestro problema, tal como se describe en Alvarez-Valdés y Tamarit /2/. En esencia, el método es el siguiente: resolver en cada nudo el problema sin tener en cuenta las restricciones sobre recursos, es decir, buscar como tiempos de realización para cada operación i una serie consecutiva de d_i tiempos tales que la suma de los β_{it} sea mínima. Este proceso es muy simple y, si no conduce a una solución posible, proporciona una cota inferior que puede ser mejorada mediante la reasignación de algunas operaciones. Para esta reasignación Ross y Soland resuelven un conjunto de problemas de knpasack para los tiempos en los que se violan las restricciones de recursos. En nuestro caso, la cota obtenida puede mejorarse mediante la resolución de problemas que podemos calificar de "pseudoknapsack", pues se trata de problemas con estructura de knapsack pero en los que el coste de reasignación de cada operación depende de las demás operaciones. En la terminología del knapsack, el valor de cada objeto no es independiente de los demás, sino que depende de la decisión que se adopte sobre ellos. La complejidad adicional en la resolución de estos subproblemas, pues no pueden usarse los algoritmos de knapsack, queda sobradamente compensada por la mejora de las cotas y la mayor rapidez en la obtención de soluciones posibles. La ramificación procede fijando ciertas operaciones a tiempos - de realización.

La estructura del algoritmo, tanto en la elección de mínimos costes como en la reasignación de operaciones, es tal que permite una simplificación cuando existe una ordenación - respecto al tiempo en los costes β_{it}. Esta deseable ordenación puede conseguirse, al menos parcialmente, mediante la introducción de un factor dependiente del tiempo mayor en magnitud que las restantes componentes del coste. Por ello en la formulación del problema se introduce en la función objetivo el factor 10t, que no lo altera y facilita la solución de (PR_β).

4.3. RESULTADOS COMPUTACIONALES

La cota inferior se obtiene, pues, relajando el problema original mediante unos multiplicadores y resolviendo los dos subproblemas resultantes. Para mejorarla se ha utilizado el método del subgradiente, realizando en cada caso 30 iteraciones.

El procedimiento se ha probado sobre una colección de problemas test similares a los del apartado 3, excepto por el hecho de que, para utilizar la versión disponible del algoritmo - de resolución de $(PR\beta)$, hemos trabajado con una sola restricción sobre los recursos.

Los resultados obtenidos aparecen en la Tabla 2, cuyas columnas son:

(1) número del problema

(2) solución proporcionada por el PERT

(3) cota obtenida mediante el método descrito

(4) solución óptima del problema.

La distancia relativa de la cota a la solución óptima del problema es, en promedio, del 6.99%, lo que indica que podría ser utilizada de forma eficiente en un algoritmo de branch & bound.

La calidad de la cota obtenida no está afectada, como ya ocurriera en 3.6, por la estructura de las relaciones de precedencia, pero la distancia de la cota al óptimo sufre un ligero aumento cuando el problema está más fuertemente restringido. Así por ejemplo en la Tabla 2 observamos que para problemas poco restringidos, como 1,2,4 y 13 las distancias relativas de la cota al óptimo se encuentran entre un 3 y un 9%, mientras que para problemas muy restringidos como 10,11,14 y 19 la distancia está entre el 7 y el 18%. Este aumento no se produce en las cotas obtenidas mediante la introducción de los planos de corte del apartado 3, lo que parece hacerlas más aconsejables para el estudio de problemas fuertemente restringidos.

5. BIBLIOGRAFIA.

/1/ R. ALVAREZ-VALDES, J.M.TAMARIT: "Una generalización del PERT: introducción de costes sobre las operaciones". Actas de la XIII Reunión Nacional de la S.E.I.O.E.I., Valladolid 1982.

/2/ R. ALVAREZ-VALDES, J.M.TAMARIT: "Un algoritmo de asignación para un problema de secuenciación con restricciones". Actas de la XIV Raunión de la S.E.I.O.E.I., Granada 1984.

/3/ E. BALAS: "Machine Sequencing: Disjunctive Graphs and Degree-Constrained Networks". Naval Research Logistic Quarterly, Vol. 17, 1970, pag. 1-10.

/4/ E. BALAS: "Project Scheduling with Resource Constraints". Applications of Mathematical Programming Techniques. Ed by E.M.L. Beale; American Elsevier, New-York 1970.

/5/ G.E. BENNINGTON, L.F. MCGINNIS: "A critique of Project Planning with Constrained Resources". Symposium on the Theory of Scheduling and its Applications, North Carolina State University, Raleigh, Ed. by S.E. Elmaghraby. Springer-Verlag, New-York, 1973.

/6/ E.W. DAVIS, G.E. HEIDORN: "An Algorithm for Optimal Project Scheduling under Multiple Constraints". Management Science, Vol. 17, 1971, pag. B803-816.

/7/ M.L. FISHER: "Optimal Solution of Scheduling Problems using Lagrangean Multipliers. Part I". Operations Research, Vol. 21, 1973, pag. 1114-1127.

/8/ M.L. FISHER: "Optimal Solution of Scheduling Problems using Lagrangean Multipliers. Part II". Symposium on the Theory of Scheduling and its Applications, North Carolina State University, Raleigh. Ed. by S.E. Elmaghraby. Springer-Verlag, New York, 1973.

/9/ S. GORENSTEIN: "An Algorithm for Project (Job) Scheduling with Resource Constraints". Operations Research, Vol. 20, 1972, pag. 835-850.

/10/ A.L. GUTJAHR, G.L. NEMHAUSER: "An algorithm for the Line-Balancing Problem", Management Science, November 1964.

/11/ J.H. PATTERSON, G.W. ROTH: "Scheduling a Project under Multiple Resource Constraints: A Zero Programming Approach". AIEE Transactions, Vol. 8, 1976, pag. 449-455.

/12/ A.A. PRITSKERS, L.J. WATERS, P.M. WOLFE: "Multiproject Scheduling with Limited Resources: A Zero-one approach". Management Science, Vol. 16, 1969, pag. 93-108.

/13/ G.T. ROSS, R.M. SOLAND: "A Branch & Bound Algorithm for the Generalized Assignment Problem". Mathematical Programming, Vol.8 1975, pag. 91-103.

/14/ J.P. STINSON, E.W. DAVIS, B.H. KHUMAWALA: "Multiple Resource-Constrained Scheduling using Branch & Bound". AIEE Transactions, vol. 10, 1978, pag. 252-259.

/15/ F.B. TALBOT, J.H. PATTERSON: "A Efficient Integer Programming Algorithm for Solving Resource-Constrained Scheduling Problems" Management Science, Vol. 24, 1978, pag. 1163-1174.