UN ALGORITMO DE ENUMERACIÓN PARA EL PROBLEMA KNAPSACK

F. RUIZ DE FRANCISCO Y J. LARRAÑETA

En este trabajo se presenta un algoritmo de resolución del problema de Knapsack basado en el análisis de una secuencia de problemas, derivados del original, desarrollando un criterio que relaciona la admisibilidad entre ellos. Este algoritmo es de enumeración implícita; examinando sucesivamente soluciones lexicográficamente ordenadas con criterios de dominancia y optimalidad. Mediante experiencias computacionales se comparan los resultados de este algoritmo con otros bien conocidos.

1. INTRODUCCION

El programa lineal de una restricción

$$\begin{aligned} \max \quad & \sum_{j=1}^{N} c_j x_j \\ \text{s.a.} \quad & \sum_{j=1}^{N} a_j x_j \leq b \\ & x_j \geq 0, \text{ entero} \end{aligned} \tag{1}$$

donde b, c_j y a_j son todos ellos números enteros positivos, se conoce como problema Knapsack. Una amplia revisión sobre el tema se encuentra en /1/.

El problema (1) puede reformularse como

$$\begin{aligned} & \text{maximizar } z \\ & \text{sujeto a } \sum_{j=1}^{N} c_j x_j \geq z \\ & \qquad \sum_{j=1}^{N} a_j x_j \leq b \\ & \qquad x_j \geq 0, \text{ entero} \end{aligned} \tag{2}$$

Siendo $\rho_j = c_j/a_j$, y renumerando las variables si fuera necesario, se asume que $\rho_i \leq \rho_j$ para $i \leq j$ con $\rho_o = 0$.

2. TERMINOLOGIA Y NOTACION

Aplicando secuencialmente la técnica de eliminación de variables de Fourier-Motzkin a las restricciones de (2) obtenemos el sistema equivalente de ecuaciones,

$$\begin{aligned} & \sum_{k=p}^{N} c_k^p x_k \geq z^p \\ & \sum_{k=p}^{N} a_k x_k \leq b \end{aligned} \tag{3}$$

con p desde 1 hasta N. Nótese que el sistema (3) tiene 2N ecuaciones, y que los coeficientes que en él intervienen toman los valores:

$$c_k^{p+1} = c_k^p a_p - a_k c_p^p$$

$$z^{p+1} = z^p a_p - b \ c_p^p$$

con

$$c_k^1 = c_k \qquad \text{y} \qquad z^1 = z$$

En el sistema (3), de 2N ecuaciones, las dos primeras (en las que p=1) son las originales y las restantes relajaciones sucesivas, puesto que las soluciones x_j han de ser enteros no negativos.

Reescribiendo los coeficientes c_k^p y z^p en términos de los originales de (1), obtenemos que

$$c_k^p = (c_k a_{p-1} - a_k c_{p-1}) \cdot a_1 \cdot a_2 \cdot \ \dots \ \cdot a_{p-2}$$

$$z^p = (z \ a_{p-1} - b \ c_{p-1}) \cdot a_1 \cdot a_2 \cdot \ \dots \ \cdot a_{p-2}$$

Por tanto, el sistema (3) es a su vez equivalente a

- F. Ruiz de Francisco. Escuela Superior de Ingenieros Industriales. Departamento de Organización de la Producción. Escuela Técnica Superior de Ingenieros Industriales. Av. Reina Mercedes, s/n. Sevilla.

- Article rebut al juny 1981.

$$\sum_{k=p}^{N} (\rho_k - \rho_{p-1}) a_k x_k + b \rho_{p-1} \geqslant z$$

$$\sum_{k=p}^{N} a_k x_k \leqslant b \qquad (4)$$

para p = 1,2, ..., N.

En base a estas relaciones definimos los siguientes conceptos.

- Un par (z^o, X^o), con $X^o = (x_1^o, x_2^o, \ldots, x_N^o)$ es admisible si satisface las restricciones de (1). La solución óptima del problema Knapsack (1) se denota por (z^*, X^*).
- Un par (z^o, X^o) es q-parcialmente admisible si satisface las ecuaciones de (4) en las que $p \geqslant q$.
- Una extensión de un par (z^o, X^o) q-parcialmente admisible es cualquier par (z^o, X^e) donde las coordenadas de los vectores X^e y X^o son idénticas desde la coordenada q hasta la N.
- Para cualquier par (z^o, X^o), el resto r_q se define como

$$r_q(z^o, X^o) \equiv \sum_{k=q}^{N} (\rho_k - \rho_{q-1}) a_k x_k^o + b \rho_{q-1} - z^o$$

De acuerdo con estas definiciones se observa que un par (z^o, X^o) de componentes enteros positivos es q-parcialmente admisible si y sólo si $r_q(z^o, X^o) \geqslant 0$.

Asimismo, definimos el problema q-Knapsack como

maximizar z

$$\text{sujeto a} \quad \sum_{k=q}^{N} (\rho_k - \rho_{q-1}) a_k x_k + b \rho_{q-1} \geqslant z$$

$$\sum_{k=p}^{N} a_k x_k \leqslant b \qquad (5)$$

$$x_q, x_{q+1}, \ldots, x_N \geqslant 0 \text{ entero.}$$

Finalmente, empleamos $|a|$ para denotar la parte entera del número a.

3. RESULTADOS

Teorema.- Sean (z^i, X^i) y (z^j, X^j) pares óptimos para los problemas i-knapsack y j-knapsack, respectivamente. Entonces,

1°.- $i \leqslant j$ implica $z^i \leqslant z^j$;

2°.- si $\sum_{k=1}^{N} a_k x_k^i = b$, entonces $z^* = z^i$ y

$$X^* = (0, \ldots, 0, x_i^i, x_{i+1}^i, \ldots, x_N^i).$$

Argumento:

1°) Según la definición del resto, para cualquier par i-parcialmente admisible (z, X).

$$r_{i+1}(z,X) - r_i(z,X) = (\rho_i - \rho_{i-1})(b - \sum_{k=i}^{N} a_k x_k) \geqslant 0 \qquad (6)$$

dado que $\rho_i \geqslant \rho_{i-1}$ por la ordenación de los datos en (1), y

$$\sum_{k=i}^{N} a_k x_k \leqslant b$$

Como $r_i(z,X)$ representa la holgura de la primera restricción del problema i-knapsack (5), y $r_{i+1}(z,X)$ tiene el mismo significado para el problema (i+1)-knapsack, el hecho de que $r_{i+1}(z,X) \geqslant r_i(z,X)$ implica que z^i es una cota inferior de z^{i+1}, soluciones respectivamente óptimas de ambos problemas. Por inducción,

$$z^i \leqslant z^j \quad \text{si} \quad i \leqslant j.$$

2°) Según (6),

$$r_i(z,X) - r_{i-1}(z,X) = (\rho_{i-1} - \rho_{i-2})(b - \sum_{k=i}^{N} a_k x_k - a_{i-1} x_{i-1})$$

Para el par (z^*, X^*) tenemos que

a) $x_{i-1}^* = 0$

b) $\sum_{k=i}^{N} a_k x_k^* = b$

Por tanto $r_i(z^*, X^*) - r_{i-1}(z^*, X^*) = 0$. Por inducción

$$r_i(z^*, X^*) = r_1(z^*, X^*)$$

Además, $r_i(z^*, X^*) = r_i(z^i, X^i)$ puesto que ---

$z^* = z^i$ y las coordenadas de los vectores X^* y X^i son idénticas desde la i hasta la N.

Teniendo en cuenta que (z^i, X^i) es la solución óptima del problema i-knapsack, (z^*, X^*) es admisible para el problema 1-knapsack, y $z^* \leqslant z^i$ de acuerdo con el apartado 1°). Luego, (z^*, X^*) es la solución óptima del problema 1-knapsack.

Corolario 1.- Sea el par (z^i, X^i) i-parcialmente admisible. Para cualquier extensión X^o de X^i, el par (z^i, X^o) es (i-1)-parcialmente admisible siendo $x_j^o = x_j^i$ para todo $j \geqslant i$, si y sólo si

$$a_{i-1}\, x_{i-1}^o + r_i(z^i, X^i)/(\rho_{i-1} - \rho_{i-2}) \geqslant$$
$$\geqslant b - \sum_{k=i}^{N} a_k\, x_k \geqslant a_{i-1}\, x_{i-1}^o$$

Argumento: la primera desigualdad es equivalente a

$$r_{i-1}(z^i, X^o) \geqslant 0.$$

La segunda procede de imponer que

$$\sum_{k=i-1}^{N} a_k\, x_k^o \leqslant b. \quad //$$

Corolario 2.- Si (z^i, X^i) es i-parcialmente admisible y

$$b - \sum_{k=1}^{N} a_k\, x_k^i$$

es un múltiplo de a_{i-1}, entonces el par

$$(z^i + |r_i(z^i, X^o)|,\ X^o)$$

es 1-parcialmente admisible, siendo

$$x_j^o = \begin{cases} x_j^i & \text{para } i \leqslant j \leqslant N \\ (b - \sum_{k=i}^{N} a_k\, x_k^i)/a_{i-1} & \text{para } j = i-1 \\ 0 \text{ para } 1 \leqslant j \leqslant i-2 \end{cases}$$

Argumento: Tenemos las siguientes igualdades,

$$\begin{aligned} r_1(z^i + |r_i(z^i, X^o)|, X^o) &= r_{i-1}(z^i + |r_i(z^i, X^o)|, X^o) && (\text{pues } x_j^o = 0 \text{ para } 1 \leqslant j \leqslant i-2) \\ &= r_{i-1}(z^i, X^o) - |r_i(z^i, X^o)| && (\text{por definición de } r_i(z, X)) \\ &= r_i(z^i, X^o) - (\rho_i - \rho_{i-1})(b - \sum_{k=i-1}^{N} a_k\, x_k^o) - \\ &\quad - |r_i(z^i, X^o)| && (\text{según } (6)) \\ &= r_i(z^i, X^o) - |r_i(z^i, X^o)| \\ &\quad (\sum_{k=i-1}^{N} a_k\, x_k^o = b) \\ &\geqslant 0 // . \end{aligned}$$

4. ALGORITMO

Usando los resultados anteriores se propone un algoritmo de enumeración. El núcleo del algoritmo consiste en comprobar si, dado un valor z^o, existe un par (z^o, X^o) admisible para el problema (2). En cada iteración del algoritmo, z^o se selecciona como valor medio (entero) del intervalo cuyos extremos son una cota superior no admisible z^S y una cota inferior admisible z^I. Inicialmente, z^S se obtiene como el valor de la función objetivo del problema continuo derivado de (1), redondeado por exceso; y z^I como el valor de la función objetivo para la solución lexicográfica máxima (solución "greedy"), |2|. Al finalizar cada iteración, si se ha encontrado un par admisible (z^o, X^o), la cota inferior z^I se actualizará dándole el valor z^o. Si para z^o no existe par admisible, se actualizará este valor. Por tanto, el número de iteraciones que se requieren para que el algoritmo finalice no es superior al redondeo por exceso de $(z^S - z^I)/\log 2$.

En el proceso de búsqueda de admisibilidad de vectores lexicográficamente descendentes, se inicia cada iteración con el vector correspondiente a z^I. En cada iteración se exploran, implícitamente, los vectores lexicográficamente menores que el X^o correspondiente al último par admisible (z^I, X^o).

Los resultados de la sección anterior se aplican para realizar la enumeración implícita de vectores. Así, el cálculo del resto informa sobre si existen extensiones admisibles del vector que se está probando. El corolario 2 cuando es aplicable, reduce el número de iteraciones del algoritmo al incrementar el valor de la cota inferior z^I.

En la figura 1 se desarrolla el diagrama de flujo del algoritmo. En él, la variable --- SLAC para un determinado valor de i, recoge la holgura de la restricción

$$\sum_{k=i}^{N} a_k x_k \leqslant b$$

Con cada uno de los algoritmos se han resuelto 100 problemas con todas las combinaciones obtenidas de la siguiente forma:

1. ρ_{max} = 1,3,13
2. b = 1000,1500,2500 y un valor aleatorio entre 1000 y 2500.
3. N = 10, 20, 30, 40, 50, 60, 70 y 100.

Un total de 9600 problemas han sido resueltos por cada uno de los algoritmos propuestos por:

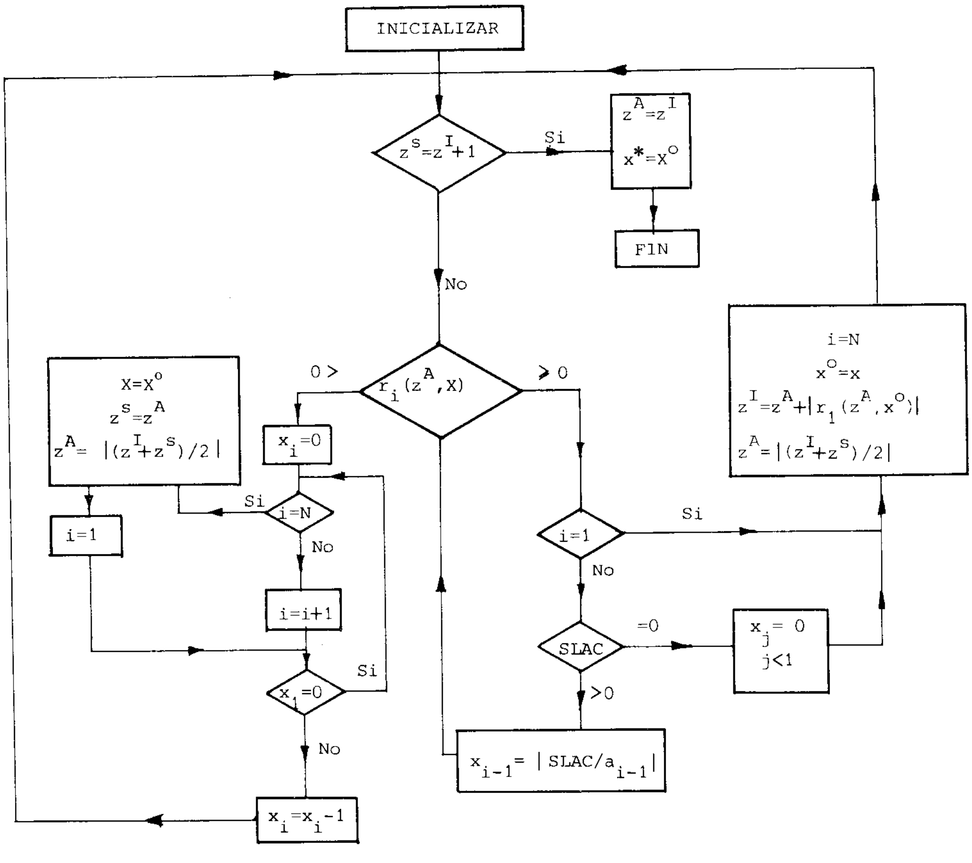


Figura 1: Diagrama de flujo del algoritmo

5. EXPERIENCIAS COMPUTACIONALES

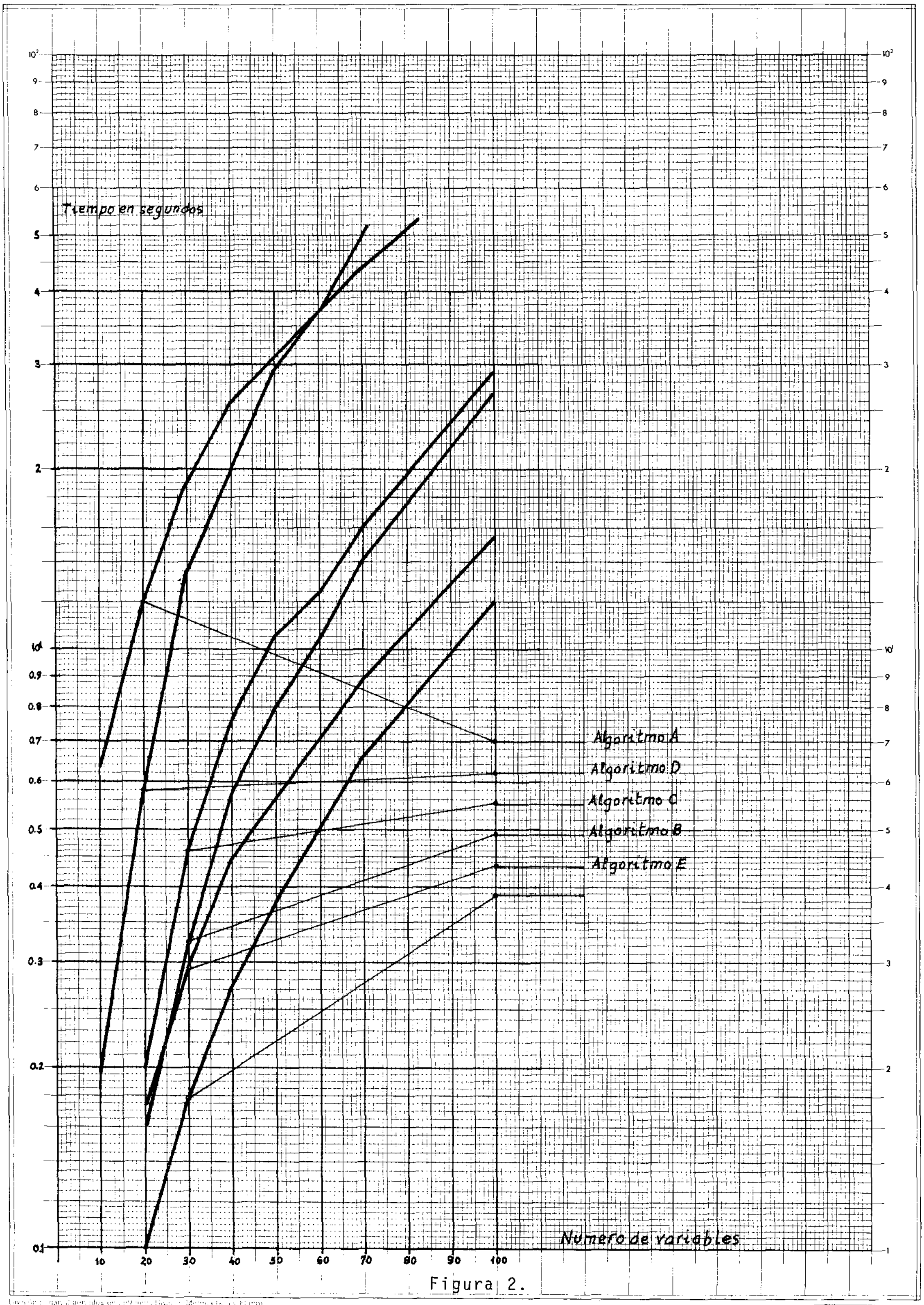
Se han realizado experiencias computacionales resolviendo problemas con el algoritmo propuesto y otros ampliamente conocidos.

Los problemas resueltos se han generado aleatoriamente con las siguientes características

a) Los coeficientes a_j y c_j(j=1,...,N) se han obtenido aleatoriamente de una distribución uniforme entre 10 y 170.
b) Para todo i ≠ j, $a_i \neq a_j$.
c) Para cada i, $\rho_i \leqslant \rho_{max}$ fijado en cada caso.

a) Algoritmo A propuesto por Gilmore y Gomory |3|.
b) Algoritmo B propuesto por Gilmore y Gomory |4|.
c) Algoritmo C propuesto por Cabot |5|.
d) Algoritmo D propuesto por Ingargiola y Korsh |6|.
6) Algoritmo E propuesto en este trabajo.

Las experiencias computacionales han sido realizadas en un ordenador H.P.21MX y un resumen de los resultados están en las tablas -- 1(N=10), 2(N=50), 3(N=100) con los tiempos-- medios en segundos.



Comparación de tiempos medios

En la figura 2 se incluyen los tiempos medios empleados por cada algoritmo en la resolución de los problemas.

6. CONCLUSIONES

Al margen del algoritmo propuesto y para problemas con 10 variables los tiempos de ejecución para los algoritmos B y C son similares. El algoritmos A es en este caso siempre más lento. Para problemas con 50 y 100 variables se incrementa la eficiencia del algoritmo A con el número de variables, al mismo tiempo que se observa como disminuye la efectividad del algoritmo D al aumentar el número de variables.

Una disminución de ρ_{max} provoca un fuerte aumento en los tiempos de ejecución de los algoritmos A y C, siendo más moderado el efecto en B y D.

Respecto al algoritmo E propuesto se demuestra que su rapidez es superior a todos los demás en más del 97% de los problemas resueltos, y sus tiempo medios de ejecución son *siempre* más pequeños que todos los demás.
Por otra parte su funcionamiento solamente está afectado por el número de variables del problema independientemente de ρ_{max} y del término independiente. Así por ejemplo, el rango de tiempos de ejecución va de 1.03 a 1.62 segundos para los 1200 problemas resueltos con 100 variables.

7. BIBLIOGRAFIA

/1/. Salkin, H., y C.A.Dekluyver (1975). "The Knapsack Problem: A Survey". Naval Research Logistics Quarterly 22, pp. 127-144

/2/. Magazine, M., Nemhauser , G.L. y Trotter, L.E. (1975). "When the Greedy solution solves a class of knapsack problems" Operations Research 23, pp. 207-217.

/3/. Gilmore, P.C. y Gomory, R.E. (1966). "The Theory and Computation of Knapsack Functions." Operations Research 14, pp. 1045-1074

/4/. Gilmore, P.C. y Gomory, R.E. (1963) "A Linear Programming Approach to the Cutting Stock Problem, Part II". Operations Research 11, pp. 863-888.

/5/. Ingargiola, G.P. y J.F. Korsh (1977) "A General Algorithm for One Dimensional Knapsack Problemas, Operations Research 25, pp 752-759