ALGORITMOS DE ASIGNACION DE RECURSOS

Miguel Angel Hinojosa Ramos, Amparo Mármol Conde
Dpto. Economía Aplicada. Universidad de Sevilla

1.– INTRODUCCION

El objeto de este trabajo es exponer algunos procedimientos para resolver problemas de asignación de recursos. En particular nos ocuparemos de problemas con variables enteras donde el criterio de decisión adecuado es el llamado criterio MAX-SUM, consistente en la maximización de la suma de las utilidades correspondientes a las distintas variables.

Aunque, en principio, trataremos los problemas de asignación como procesos de decisión secuencial, pudiendo así hacer uso de técnicas de programación dinámica, consideraremos también, procedimientos basados en otras ideas, como la asignación marginal y los multiplicadores de Lagrange.

2.– EL PROBLEMA DE LA MOCHILA

Consideraremos un problema de asignación de recursos en el que tanto la función objetivo como la restricción son lineales: el llamado problema "de la mochila", que puede interpretarse como la maximización del valor de la carga de una mochila dados los valores y los volúmenes de una serie de artículos.

La formulación matemática de este problema será:

$$\text{Max } p_o x_o + p_1 x_1 + ...+ p_N x_N$$

$$\text{s.a. } c_o x_o + c_1 x_1 + ...+ c_N x_N = V$$

$$0 \leq x_n$$

$$x_n \text{ entero } n= 0, ..., N$$

Donde V es el volumen de la mochila. Cada unidad del artículo n ocupa un volumen C_n (entero positivo) y reporta un beneficio p_n. Permitiremos que la mochila quede parcialmente vacia considerando un artículo "0" tal que cada unidad ocupa una unidad de volumen (C_o= 1) y no produce beneficios.

La restricción de que las variables, sean enteras es fundamental, pues de no serlo se comenzaría llenando con el producto tal que la razón p_n / C_n sea máxima (mayor beneficio y mínimo volumen) hasta agotar las disponibilidades de dicho artículo, a continuación con el que de la segunda mayor proporción y así sucesivamente.

Suponiendo que se van empaquetando los artículos en la mochila uno tras otro, convertiremos el modelo de asignación en un proceso de decisión secuencial. Si se han puesto en la mochila

una serie de artículos y quedan j unidades de volumen, al ser las funciones lineales, el problema de llenar el resto de la mochila maximizando el beneficio es independiente del contenido anterior. Podemos establecer:

MAX (j) = máximo beneficio que se obtiene al llenar una mochila que dispone de j unidades de volumen.

Teniendo en cuenta que poner una unidad del artículo n en una mochila de volumen j da lugar a un beneficio p_n y reduce el volumen disponible a $j-C_n$, se obtiene:

$$MAX(j) = \max \left[p_n + MAX(j-C_n) \right]_{n / c_n \leq j} \quad j = 1,.....,V$$

Educación funcional que permitirá resolver nuestro problema de manera recursiva. En realidad se resolverá una familia de problemas, correspondientes a los volúmenes parciales. Su resolución conlleva V(N + 1) evaluaciones ((N+1) para cada j=1,.....,V).

Este problema puede tratarse también como un problema de camino más largo de la siguiente manera: se crean),....., V nodos y una serie de arcos (i, i + C_n) de longitud p_n (i = 1,.....,V, n = 1,.....,N, i + $C_n \leq K$). De este modo MAX (j) (máximo beneficio obtenido al empaquetar "j" unidades de volumen disponibles) será la longitud del camino más largo desde el nodo «0» al nodo "j" en el grafo que hemos creado.

Desde este punto de vista podemos resolver el problema mediante el siguiente procedimiento:

Algoritmo 1

1. Sea MAX (j) = 0 j = 0,....,V
2. Hacer para i-0,...., V - 1
3. Hacer para n = 0,....,N
    4. $j = i + C_n$

Si j<V entonces $MAX(j) = \max MAX(j), MAX(i) + p_n$

Este procedimiento que es un método para calcular el camino más largo en un grafo, puede mejorarse teniendo en cuenta lo siguiente: hasta ahora el valor de los artículos de la mochila no se ve afectada por el orden en que se coloquen. Vamos a restringir nuestra atención a los caminos del grafo que hemos creado en el procedimiento anterior comenzando en el nodo 0 cuyos arcos iniciales corresponden al empaquetado del artículo N seguidos por los arcos que corresponde al artículo N-1 y así sucesivamente terminando con los arcos que corresponden al artículo 0. En este sentido los artículos se empaquetan en orden decreciente.

Definimos COTA (j) como el artículo con menor índice que es óptimo para empaquetar en una mochila con j unidades de volumen: COTA (j) = min n/MAX (j) = p_n + MAX (j - C_n) $n \geq 1$.

Además haremos COTA (0) = N, pues en principío no tenemos información para acotar los productos a introducir en la mochila. De esta manera será óptimo colocar el artículo COTA (j) en una mochila de volumen j. La siguiente observación es fundamental para deducir el procedimiento que estamos perfilando: Comparemos COTA (j) con COTA (j - c_n): si disponemos de una mochila de volumen j habrá un conjunto de artículos para los que se obtendrá el óptimo que, por otro lado, contiene al conjunto de artículos para los que se obtendrá el óptimo en un nodo anterior (una mochila de volumen j-c_n); de aquí que COTA (j) que es el índice más bajo en el conjunto mayor no es mayor que COTA (j- c_n) que es el artículo con índice más bajo del conjunto menor, es decir:

$$\forall \quad j > 1 \quad COTA(j - Cn) > COTA(j).$$

A continuación modificamos el procedimiento anterior para hacer uso de este resultado:

Algoritmo 2

1. MAX (j) = 0 para j = 0,.....,V
   COTA (j) = N para j = 0,.....,V
2. Hacer para i = 0,....., V-1
3. Hacer para n = 0,....., COTA (i)
4. Si $(i + C_n) < V$ ir al paso 5

Si no: MAX $(i + C_n)$ = max MAX $(i + C_n)$, MAX (i) + p_n

Si ha habido mejora estricta (el máximo se alcanza en MAX (i) + pn) actualizar:
COTA (i + Cn) = n
Si se da la igualdad 9 MAX (i + Cn) = MAX (i) + pn) actualizar: COTA (i + Cn) =
= min (COTA (i + Cn), n)

5. *Continuar*

Este procedimiento supone una mejora considerable del anterior pues gran parte, de los arcos del grafo pueden eliminarse en cuanto a cálculos se refiere, ya que solo en el nodo 0 (COTA (0)=0) tendremos que evaluar todas las posibilidades de llenado, así probablemente eliminaremos ramas enteras del grafo de la siguiente manera: desde un nodo genérico i en el que tendremos que evaluar COTA (i) artículos en orden creciente, la primera vez que se produzca una mejora estricta actualizamos COTA $(i + C_n)$ con un valor menor que COTA (i) con lo que eliminamos todas las ramas del grafo que parten del nodo $i + C_n$ considerando los productos COTA $(i + C_n) + 1$ hasta N.

3. *Asignación no lineal de recursos*

a) Asignación de un solo recurso:

Consideremos un problema más general de asignación de recursos:

$$\text{Max } p_1 (x_1) + p_2 (x_2) + ... + p_N (X_N)$$

$$\text{s.a. } C_1 (x_1) + C_2 (x_2) + ... + C_N (x_N) \leq R$$

$$0 \leq x_n < b_n$$

$$x_n \text{ enteros } n = 1,...., N$$

El problema puede interpretarse como la asignación de R unidades de recurso disponibles a la fabricación de N productos diferentes de manera que se obtenga el máximo beneficio. Los productos pueden fabricarse solo en cantidades enteras y la producción de x_n unidades del producto n-esimo consume una cantidad no negativa de recurso que consideramos entera, $c_n (x_n)$, y reporta un beneficio $p_n (x_n)$. El decisor no tiene por que asignar todo el recurso disponible, pero no puede fabricar más de b_n unidades del producto n.

Supongamos que se fabrica en primer lugar el producto N, luego el producto N-1 y asi sucesivamente hasta el 1. Si se han fabricado ya varias cantidades de los productos n + 1 hasta N para lo que se ha gastado cierta cantidad de producto y quedan «y» unidades ($0 \leq y \leq k$), estas deberán ahora asignarse a la fabricación de los artículos 1,...., N.

Llamaremos MAX (n,y) = Beneficio total máximo de asignar "y" unidades de recurso a la fabricación de los productos 1,....,N.

Hemos enmarcado el problema en una familia de N (R+1) problemas de optimización relacionados entre si, consistentes en la determinación de MAX (n,y), n= 1,2,....., N, y= 0,1,....., R. Por definición MAX (N,R) es el máximo beneficio para nuestro problema de asignación.

La siguiente ecuación funcional proporciona un proceso recursivo para resolver el problema:

$$MAX\ (n,y) = Max\ \ p_n\ (x) + MAX\ (n\text{-}1,\ y\text{-}C_n\ (x))\ \ ,\ n>1$$

s.a. x factible

$$MAX\ (1,y) = Max\ \ p_1\ (x)$$

s.a. x factible

Donde x factible significa que verificando las restricciones del problema, se cumple $C_n\ (x) \leq y$.

El volumen de cálculos para este modelo crece considerablemente en relación con el modelo de la mochila, pues por cada uno de los N (R + 1) problemas que hay que resolver tendremos que realizar diversas evaluaciones.

Si consideramos una representación del problema mediante grafos, estableceremos nodos (n,y) y un nodo adicional (0,0). Los arcos de la red se definen de forma que la longitud del camino más largo del nodo (0,0) al nodo (n,y) es el mayor beneficio posible, cuando se asignan ´y´unidades del recurso a la producción de los productos 1 hasta n.

Si se satisfacen las condiciones:

$$x_n\ \varepsilon\ (\ 0,1,.....,\ b_n)$$

$$y\ +\ C_n\ (x_n) \leq R$$

decimos que x_n es un nivel de producción factible para el producto n en el nodo (n-1,y).

Para cada nodo (n-1,y) y cada x_n factible para este nodo, se crea un arco de longitud $p_n\ (x_n)$ que parte del nodo (n-1,y) y termina en el nodo (n, y + $C_n\ (x_n)$). Con esta construcción, el camino más largo del nodo (0,0) al nodo (n,y) corresponde al mayor beneficio posible al asignar exactamente ´y´unidades de recurso a la fabricación de los productos 1 hasta n-1.

De todas formas, no es necesario consumir todo el recurso. Para permitir que alguno no se use, se crean arcos ficticios de longitud 0 que parten de (n,y-1) y terminan en (n,y), $1 \leq n \leq N$, $1 \leq y \leq R$.

El procedimiento que consideraremos hace una comparación para cada arco que parte del nodo (n,y). Para el arco ficticio, reemplaza MAX (n, y+1) por MAX (m,y) cuando este es mayor. Para cualquier otro arco que parte del nodo (n,y) se reemplaza MAX (n+1, y+ $C_{n+1}\ (x)$ por MAX (n,y) + p_{n+1} (x) cuando el segundo es mayor. Se realizan estas comparaciones, en primer lugar, para los arcos que parten de (0,0), después para los que parten de (1,y) con ´y´creciente, después para los de (2,y), y así sucesivamente.

Este procedimiento mejora el método recursivo anterior pues algunos arcos serán pronto eliminados. Observese que MAX (n,y) ≥ MAX (n,y-1), pues la asignación que alcanza MAX (n,y-1) permanece factible cuando el nivel de recurso se incrementa hasta y.

Además adoptaremos el siguiente criterio:

Si MAX (n,y) = MAX (n,y-1), no crearemos arcos que partan del nodo (n,y) excepto los arcos ficticios.

Se puede probar que con este criterio no se eliminan todos los caminos más largos del nodo (0,0) a ningún nodo.

A continuación se describe el procedimiento que incorpora este criterio.

Algoritmo 3

1. MAX (0,0) =), MAX (n,y) = $-\infty$ en otro caso.
   Para x = 0,1,...., b_1 MAX (1, c_1 (x))= max (MAX (1, c_1, (x)), p_1 (x))
2. Hacer para n = 1,....,N
3. Hacer para y = 0,....,R
   4. Si y ≥ 1 y MAX (n,y) ≤ MAX (n, y-1) entonces:
      MAX (n,y) = MAX (n, y-1)
      ir al paso 7.
   Si n= N ir al paso 7.

5. En otro caso, hacer para x=0,....,b

6. $z = y + c_{n+1}(x)$.

Si $z \leq k$ entonces.

$$MAX(n+1,z) = \max(MAX(n+1,z), MAX(n,y) + p_{n+1}(x))$$

7. Continuar.

La ventaja de este procedimiento es que asocia arcos con nodos cuyos caminos más largos ya se han calculado. Aunque no reduce la cota teórica de cálculos, este procedimiento es más rápido para la mayoría de los problemas.

b) Asignación de varios recursos

Los modelos considerados hasta ahora tratan de la asignación de un único recurso para la fabricación de distintas cantidades de una serie de productos. En muchos casos son varios los recursos que intervienen en la fabricación de los productos. Generalizaremos a estos casos la formulación y resolución del problema anterior.

Esta generalización no supone dificultades conceptuales pero sí puede crear problemas de computación. Consideremos el modelo con dos recursos:

$$\text{Max } p_1(x_1) + p_2(x_2) + \ldots\ldots + p_N(x_N)$$

$$\text{s.a.: } C_1(x_1) + C_2(x_2) + \ldots\ldots + C_N(x_N) \leq R_1$$

$$d_1(x_1) + d_2(x_2) + \ldots\ldots + d_N(x_N) \leq R_2$$

$$0 \leq x_n \leq b_n$$

$$x_n \text{ entero} \quad n = 1,\ldots\ldots,N$$

Hay disponibles R_1 y R_2 unidades de los recursos 1 y 2 respectivamente. La fabricación de x unidades del producto n consume $d_n(x)$ unidades del recurso 2 y $C_n(x)$ unidades del recurso 1. Estas funciones de consumo se supone que toman valores no negativos con $Cn(0) = 0 = dn(0)\ \forall\ n$.

Supongamos que se han establecido las cantidades a fabricar de los productos n+1 a N con lo que se han utilizado parte de los dos recursos y quedan 'y' unidades del recurso 1 y 'z' unidades del recurso 2. Sin conocer los otros niveles de producción puede hacerse la asignación de estas unidades a la producción de los productos 1, ...,n haciendo uso de la siguiente ecuación funcional:

$$\text{Max } (n,y,z) = \text{Max } (p_n(x_n) + \text{Max } (n-1,\ y - C_n(x),\ z - d_n(x))$$

$$C_n(x) \leq y$$

$$d_n(x) \leq z$$

El procedimiento recursivo utilizado en el problema de un solo recurso puede trasladarse a este otro problema pero requiere ahora la resolución de $N(R_1 + 1)(R_2 + 1)$ problemas creciendo considerablemente los cálculos. Cuando se introducen nuevos recursos se observa que el volumen de cálculo crece de manera exponencial y la estructura del problema resulta en ocasiones demasiado grande.

Para salir al paso de este inconveniente se propone el siguiente método:

Consideremos un modelo de asignación con dos recursos e incorporemos la restricción del segundo recurso a la función objetivo de la siguiente manera:

$$\text{Max} \quad p_1(x_1) + \ldots\ldots + p_n(x_n) - \lambda \ (d_1(x_1) + \ldots + d_N(x_N))$$

$$\text{s.a.: } C_1(x_1) + \ldots + C_N(x_N) \leq R_1$$

$$0 \leq x_n \leq b_n$$

$$x_n \text{ enteros } n = 1, \ldots\ldots, N$$

(A λ se le llama «multiplicador de Lagrange»).

Observese que el número de unidades disponibles del segundo recurso, R_2, no aparece en la formulación del problema y que la recompensa por producir x unidades del producto n será $p_n(x) - \lambda d_n(x)$. De aquí que para un λ fijo el problema puede resolverse por el método anterior para un solo recurso.

Sea X el conjunto de N-uplas que satisfacen las restricciones del problema anterior (obsérvese que hay un número finito de ellas). Sea $X^\lambda \varepsilon X$ la N-upla que maximiza la función objetiva del problema anterior para λ fijo. Sean también $p(x) = p_1(x_1) + \ldots + p_N(x_N)$.

$$d(x) = d_1(x_1) + \ldots + d_N(x_N).$$

Se cumple entonces:

1) $p(x^\lambda) = \text{Max}\ (p(x)/\ x\varepsilon\ x;\ d(x) \leq d(x^\lambda))$

2) $d(x^\alpha) \geq d(x^\beta) \qquad \alpha < \beta$

La primera expresión indica que $p(x^\lambda)$ es la máxima ganancia que se obtiene si se consumen $d(x^\lambda)$ o menos unidades del segundo recurso. De aquí que si conseguimos obtener un valor de λ que satisfaga $d(x^\lambda) = R_2$ habremos encontrado una solución del problema.

La segunda expresión indica que $d(x^\lambda)$ es una función no decreciente de λ, lo que apunta a métodos para llegar a un valor de λ que satisfaga $d(x^\lambda) = R_2$.

Examinemos el problema para $\lambda = 0$ y obsérvese que $p(x^o)$ es el mayor beneficio que se obtiene en $x \varepsilon X$. Si $d(x^o) \leq R_2$ entonces tenemos ya una solución del problema y la segunda restricción es redundante.

A la vista de las expresiones (1) y (2) el problema de hallar el λ adecuado podría resolverse por el método de bisección que consiste en lo siguiente:

1. Resolver el problema con valores 'b' y 'c' para λ que aseguren $d(x^b) > R_2$ y $d(x^c) < R_2$.
2. Resolver el problema para $\lambda = (b+c)/2$.
   Si $d(x^\lambda) = R_2$. FIN.
3. Si $d(x^\lambda) < R_2$ hacer $c = \lambda$
   Ir a 2.

   En otro caso hacer $b = \lambda$

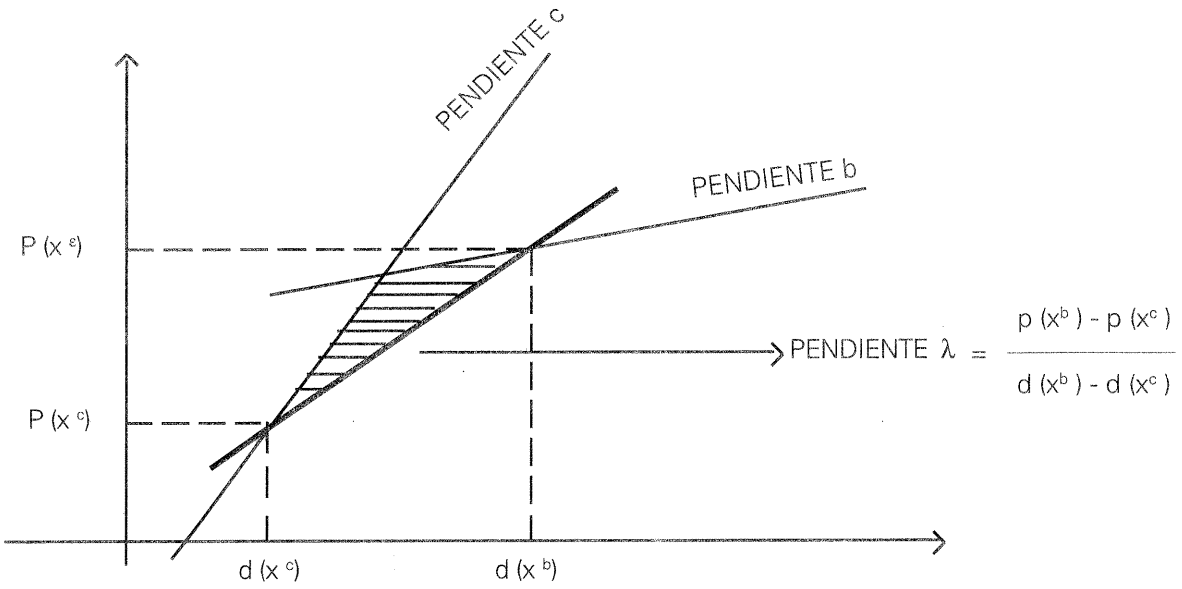
La bisección, en general, no garantiza la terminación finita del algoritmo, salvo cuando el segundo recurso contribuye con rendimientos marginales decrecientes estrictamente. Para obtener un algoritmo más eficiente consideraremos también el siguiente método:

Supongamos que el problema se ha resuelto para valores b y c de λ que satisfacen:

$$d(x^b) > R_2 > d(x^c)$$

Si hacemos una representación del problema tomando como ejes coordenados d (x) y p (x), λ representa la pendiente de la recta que une los puntos correspondientes a 'b' y 'c'. La función objetivo $p(x) - \lambda d(x)$ es constante a lo largo de esta recta y la nueva solución x^λ verifica necesariamente:

(*) $$p(x^\lambda) - \lambda d(x^\lambda) \geq p(x^c) - \lambda d(x^c)$$



Como x^b y x^c maximizan la función objetivo del problema cuando el multiplicador es igual a 'b' y 'c' respectivamente, cada elemento $x \varepsilon X$ debe tener asociado (d (x), p (x)) que esté sobre o por debajo de las líneas punteadas de pendientes 'b' y 'c'. De la misma forma, como x^b y $x^c \varepsilon X$, el elemento x^λ debe tener (d (x^λ), p (x^λ)) sobre o por encima de la línea de pendiente λ, es decir, este punto estará en la región sombreada.

El procedimiento consistirá:

1. Resolver el problema para valores 'b' y 'c' de λ que aseguren d (x^b) > R_2 y d (x^c) < R_2.

2. Resolver con λ = (p (x^b) - p (x^c)) / (d (x^b) - d (x^c)).
   Si p (x^λ) - λ d (x^λ) = p (x^c) - λ d (x^c) entonces

3. Si d (x^λ) < R_2 entonces c = λ
   Ir a 2
   En otro caso b = λ

La ventaja del método de actualización de la pendiente sobre el de bisección es que garantiza la finalización en un número finito de pasos; la ventaja de la bisección sobre la actualización de la pendiente es que garantiza la reducción a la mitad del intervalo (b,c). Consideraremos entonces un procedimiento que combina las ventajas de ambos métodos: Si d (x^b) - d (x^c) se reduce en alguna iteración utilizaremos el método de bisección en la siguiente, si no actualizaremos la pendiente.

4. *Rendimiento marginal decreciente*

Consideremos una función g definida sobre los enteros no negativos. Se define su diferencia primera Δ g (x), como Δ g (x) = g (x+1) - g (x), x = 0,1,.....

La función g es cóncava si su diferencia primera es no creciente, es decir, Δ g (x) $\geq$ Δ g (x=+1), x= 0,1,... (*).

Interpretamos g (x) como el beneficio de poseer x unidades de un determinado artículo, con lo que Δ g(x) es el beneficio marginal de la siguiente unidad. Una función que satisface (*) se dice también que posee rendimientos marginales decrecientes, pues el beneficio marginal de una unidad no es mayor que el de la anterior.

Estudiaremos el siguiente problema bajo el supuesto de que las funciones p_n tienen rendimientos marginales decrecientes.

$$\text{Max} \quad \left[p_o(x_o) + p_1(x_1) + \ldots + p_N(x_N) \right]$$

$$\text{s.a.:} \quad x_o + x_1 + \ldots + x_N = R$$

$$0 \leq x_n \leq b_n$$

$$x_n \text{ entero} \qquad \text{para } n = 0, \ldots, N$$

Suponiendo $p_o(x) = 0$ para todo x, la variable 0 hace el papel de variable de holgura.

En la práctica aparecen frecuentemente modelos de asignación con esta característica. Supongamos, por ejemplo, que un distribuidor dispone de R unidades de un producto perecedero que quiere distribuir entre N puntos de venta, de forma que se maximicen las ventas y que la cantidad de productos estropeado sea mínima. Se estima que en el envío de x unidades al punto n da lugar a unas ventas $p_n(x)$. Es razonable suponer que cada unidad adicional que envíe tiene una probabilidad más alta de no venderse. Es decir, las ventas decrecen marginalmente en x.

Los algoritmos recursivos basados en la Programación Dinámica pueden resolver este problema, pero las llamadas técnicas de análisis marginal que consideraremos a continuación resultarán más rápidas, e incluso más flexibles para su tratamiento.

Dada una asignación factible $X = (x_o, \ldots, x_N)$ el Análisis Marginal plantea dos cuestiones:

a) Sería conveniente incrementar en una unidad la producción de algún artículo y disminuir en una unidad la producción de otro?

b) Si es así, cuáles serían estos artículos?

$$\text{Sean} \quad I(x) = \max_{x_n < R} \left[p_n(x_n) \right]$$

$$D(x) = \min_{x_m > 0} \left[p_m(x_m - 1) \right]$$

Observemos que I(x) es el mayor incremento en el beneficio que se obtiene al aumentar algún x_n en una unidad, y D (x) la menor reducción en el beneficio al disminuir algún x_m en una unidad.

El análisis marginal calcula I (x) y D (x) y comprueba si D (x) < I (x); si es así se ejecuta la acción correspondiente de incrementar y reducir respectivamente x_n y x_m y se repite el proceso. Es decir, comenzando con un asignación factible repite el siguiente paso:

Si $I(x) \leq D(x)$ entonces FIN

En otro caso, para 'n' y 'm' tales que

$$I(x) = p_n(x_n)$$
$$D(x) = p_m(x_{m-1})$$

Incrementar x_n en una unidad y reducir x_m en una unidad.

Cada iteración incrementa el beneficio total y conserva la factibilidad. Como el número de asignaciones factibles es finito, el análisis marginal conseguirá la asignación óptima en un número finito de pasos. Las iteraciones terminarán cuando se localice una asignación X^* que verifique $I(x^*) \leq D(x^*)$.

En cada iteración debe calcularse I (x), que es la mayor diferencia, lo que puede hacerse en N comparaciones. Puede demostrarse además, que si una variable aumenta (o disminuye) en una iteración dada, no volverá a disminuir (aumentar) en adelante, de esta forma, la variable x_n que ha aumentado en una iteración puede eliminarse del cálculo de D (x). De la misma forma, la variable x_m que haya disminuido puede eliminarse del cálculo de I (x).

Otra consecuencia de esta propiedad es que puesto que en cada iteración intermedia disminuye exactamente una variable x_n cuyo valor está por encima de x_n^* y aumenta exactamente una x_m cuyo valor está por debajo x_m^*, el número de iteraciones intermedias será precisamente

$$\sum (x_n - x_n^*)^+, \qquad \text{con } z^+ = \max\ 0, z$$

esta suma vale como mucho K, por tanto el análisis marginal hallará la solución óptima en K + 1 iteraciones como máximo.

Para comparar el análisis marginal con el método recursivo suponemos que iniciamos el análisis marginal con la asignación factible (R, 0,....,0), donde solo la variable "de holgura" tiene un nivel de producción (por encima del óptimo. Los otros niveles de producción) se incrementarán solo si esto incrementa el beneficio. El análisis marginal rebaja x_o en cada iteración e incrementa en una unidad la producción del producto cuya contribución marginal al beneficio es mayor. La cantidad D (x) es inicialmente 0 y se queda en 0 durante todo el proceso. Por tanto no es necesario calcular D (x) en ninguna iteración. El cálculo de I (x) de la manera más "inocente" conlleva

la comparación de N números. Con un máximo de R+1 pasos, el número de cantidades comparadas en esta implementación de análisis marginal es como máximo N (R+1). El número de cantidades comparadas en el método recursivo es exactamente de N (R+1) (R+2) / 2 que es mayor que el factor (R+2) / 2.

Además el método de asignación marginal puede iniciarse en cualquier asignación factible. En particular puede iniciarse con una asignación que le parezca bien al decisor, mientras que el método recursivo no puede hacer uso de esta información.

Incluso puede intentarse aplicar la asignación márginal cuando algunas de las funciones recompensa no son cóncavas, en estos casos se consideraría la menor mayorante cóncava y se resolverá el problema llegando en algunos casos a la solución exacta y en otros a una aproximación de la solución, que en muchas ocasiones puede ser suficiente para nuestros propósitos, sobre todo cuando la alternativa es un volumen muy superior de cálculos y gastos.

BIBLIOGRAFIA

Denardo, E. V.: " Dynamics Programming. Models and Applications". Prentice - Hall, 1982.

Dudzinski, K.; Walukiewicz, S.: " exact Methods for the Knapsack Problem and its Generalizations". European Journal of Operational Research 20 (1987).

Minoux, M: "Mathematical Programming. Theory and Algorithms". John Wiley and Sons, 1986.

Vidal, R.: " A Graphical method to solve a Family of Allocation Problems". European Journal of Operational Research 17 (1984).