UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE QUIMICA

APLICACION DE LA PROGRAMACION DISCRETA AL ANALISIS DE INVERSIONES

T E S I S
QUE PARA OBTENER EL TITULO DE
INGENIERO QUIMICO
P R E S E N T A

ARMANDO RAFAEL ZAPIEN VILLAVICENCIO

MEXICO, D. F. 1970

UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FACULTAD DE QUIMICA
DEPTO. DE PASANTES Y
EXAMENES PROFESIONALES.

FORMA C

(AUTORIZACION PARA ESCRIBIR DEFINITIVAMENTE EL TEMA REVISADO)

C. Director Gral. de Servicios Escolares
Universidad Nacional Autónoma de México.
P r e s e n t e .

Me permito comunicar a usted, que el tema de T E S I S . Titulado: "APLICACION DE LA PROGRAMACION DISCRETA AL ANALISIS DE INVERSIONES."

que presenta: EL SR. ARMANDO RAFAEL ZAPIEN VILLAVICENCIO
Pasante de la Carrera de: INGENIERO QUIMICO
Fué aceptado por el Jurado nombrado para dicho examen, el cual quedó integrado en la siguiente forma:

Presidente Prof: EDUARDO ROJO Y DE REGIL
V o c a l " : ABELARDO F. PADIN Y LIMA
Secretario " : JORGE RIVERA BENITEZ
1er. Suplente : ENRIQUE JIMENEZ RUIZ
2do. Suplente : JOSE LUIS PADILLA DE ALBA.

A t e n t a m e n t e .
"POR MI RAZA HABLARA EL ESPIRITU"
Cd. Universitaria D.F., a 6 de OCTUBRE de 1970

EL JEFE DEL DEPTO. DE PASANTES
Y EXAMENES PROFESIONALES.

QUIM. JULIO TERAN Z.

A MIS PADRES

A MIS HERMANOS

1. INTRODUCCION

Uno de los problemas que con más frecuencia se presenta en una empresa es el de la selección del conjunto de proyectos que maximizan sus beneficios, sujetos a restricciones de capital, en un periodo de planeación especificado.

El objeto de este trabajo es describir, dentro de las técnicas de la programación matemática, el método denominado de ramificación y acotación progresiva aplicado al problema anterior, así como el desarrollo de un programa de computadora que resuelve un algoritmo basado en esta técnica.

Con este fin, se presenta inicialmente un basquejo de la teoría de dualidad de la programación lineal para fijar los antecedentes de la programación discreta. Posteriormente se hace una breve descripción histórica de la programación presentando además la formulación matemática de la técnica de ramificación y acotación.

Finalmente, en las conclusiones se discute la solución del problema planteado, y el uso del algoritmo mediante computadora.

1. TEORIA DE DUALIDAD DE LA PROGRAMACION LINEAL

1.1 Planteamiento, lemas y teoremas

El problema general de programación lineal puede explicarse definiendo la teoría de dualidad.

A continuación se explica, en forma somera, dicha teoría.

Dos problemas de programación lineal se llaman duales si tienen la siguiente forma:

Primal (P)		Dual (D)	
Max $Z = \underline{CX}$	(1)	min $Z' = \underline{d}'\underline{u}$	(4)
$A\,\underline{X} \leq \underline{d}$	(2)	$A'\,\underline{u} \geq \underline{c}'$	(5)
$\underline{X} \geq 0$	(3)	$\underline{u} \geq 0$	(6)

Donde

Z función objetivo

X vector de variables primales; $X = \{X_j\}$; $j = 1, \ldots, n$

u vector de variables duales; $u = \{u_i\}$; $i = 1, \ldots, m$

b vector de recursos; $b = \{b_i\}$; $i = 1,\ldots, m$

C vector de costos: $C = \{C_j\}$; $j = 1,\ldots, n$

A matriz de coeficientes de las restricciones; $A = \{a_{ij}\}$;

$i = 1,\ldots, m$ $j = 1,\ldots, n$

De lo anterior se observa que:

1. El objetivo en P es maximizar, y en D minimizar

2. En el sistema de minimización las desigualdades correspondientes a las restricciones son de la forma ($\geq$) y en el de maximización de la forma ($\leq$)

3. La matriz de coeficientes de un sistema es igual a la traspuesta del otro

4. Los términos independientes de un sistema son iguales a los coeficientes de la función objetivo del otro

5. El número de variables de un sistema, es igual al número de restricciones del otro

6. En ambos sistemas se satisfacen las restricciones de no negatividad ($\underline{X} \geq 0$ y $\underline{u} \geq 0$).

1.2 Teorema de dualidad

La importancia de la definición de los problemas duales se debe a que existe una relación entre sus soluciones factibles y óptimas. Una vez establecidas estas mediante un conjunto de lemas y teoremas, se pueden construir algoritmos de diversa naturaleza para resolver el problema de programación lineal.

Lema 1.1

Si $\underline{X}$ es una solución factible para el primal y $\underline{u}$ una solución factible para el dual, entonces:

$$\underline{C}\ \underline{X} \leq \underline{d}'\underline{u} \qquad \text{o} \qquad Z' \leq Z$$

Prueba: multiplicando $A\ \underline{X} \leq \underline{d}$ por $\underline{u}$ y $A'\ \underline{u} \geq C'$ por $\underline{X}'$:

$$\left.\begin{array}{l} \underline{u}'A\ \underline{X} \leq \underline{u}'\underline{d} \\ \underline{X}'A\ \underline{u} \geq X'\ \underline{C}' \end{array}\right\} \Rightarrow \quad \begin{array}{l} \underline{u}'A\ \underline{X} \leq \underline{d}'\underline{u} \\ \underline{u}'A\ \underline{X} \geq \underline{C}\ \underline{X} \end{array}$$

por lo tanto

$$\underline{C}\ \underline{X} \leq \underline{d}'\ \underline{u}$$

De lo anterior se concluye que cuando existen soluciones factibles tanto para el primal como para el dual, el valor de Z del problema de minimización constituye una cota superior para los valores de Z' y también el valor de Z' del problema de maximización constituye una cota inferior para los valores de Z.

Lema 1.2

Si $\underline{X}^o$ es una solución factible para el primal y si $\underline{u}^o$ es una solución factible para el dual y además $\underline{C}\ \underline{X}^o = \underline{d}\ \underline{u}^o$, entonces $\underline{X}^o$ y $\underline{u}^o$ son óptimos en sus respectivas soluciones factibles básicas.

Prueba

Por el lema 1.1

$$\underline{C}\ \underline{X} \leq \underline{d}'\ \underline{u}^o$$

se llega a

$$\underline{C}\ \underline{X} \leq \underline{C}\ \underline{X}^o$$

lo que implica que $\underline{X}^o$ es óptimo.

Igualmente para cualquier solución factible $\underline{u}$ del dual, se tiene

$$\underline{d}'\ \underline{u} \geq \underline{C}\ \underline{X}^{o} = \underline{d}'\ \underline{u}^{o}$$

lo que implica que

$$\underline{d}'\ \underline{u} \geq \underline{d}'\ \underline{u}^{o}$$

De lo anterior se deduce que $\underline{u}^{o}$ es óptimo.

Teorema 1.1 (de dualidad)

Una solución factible $\underline{X}^{o}$ del primal es óptima si, y solo si, existe una solución factible $\underline{u}^{o}$ del dual, para la cual

$$\underline{d}'\ \underline{u}^{o} = \underline{C}\ \underline{X}^{o}$$

Una solución factible $\underline{u}^{o}$ es óptima si, y solo si, existe una solución $\underline{X}^{o}$ para el primal tal que

$$\underline{C}\ \underline{X}^{o} = \underline{d}'\ \underline{u}^{o}$$

Teorema 1.2 (de existencia)

a) Un problema primal (por lo tanto el dual correspondiente) tiene una solución óptima si, y solo si, ambos tienen soluciones factibles

b) Si uno solo (dual o primal) de los problemas tiene solución factible, entonces su función objetivo no está acotada.

Pruebas

a) Si $\underline{X}^{o}$ es la solución óptima del primal, por el teorema de dualidad, el dual tiene solución factible $\underline{u}$. Considérese ahora que el primal y el dual tienen soluciones factibles $\underline{X}$ y $\underline{u}$; entonces $\underline{C}\ \underline{X}$ y $\underline{d}'\ \underline{u}$ son finitas. Por lo tanto, del lema 1, $\underline{d}'\ \underline{u}$ sirve como cota superior de la función objetivo del primal $\underline{C}X$, aunque no es necesariamente la menor cota superior.

Por lo tanto el primal tiene óptimo.

b) Sea el dual no factible, y supóngase que $\underline{X}$ es una solución factible del primal, con valor $\underline{C}\ \underline{X}$ para la función objetivo. Esta no puede ser óptima, pues si lo fuera, por el teorema de dualidad estaría garantizada la existencia de una solución factible del dual, lo cual contradice la primera suposición, por lo tanto no existe solución factible del primal que pueda ser óptima y se concluye que no está acotada.

1.3 Relación de las holguras complémentarias de los problemas primal y dual

Establece la correspondencia entre una solución óptima del primal y la correspondiente holgura de una solución óptima del dual.

Considérese

Primal	Dual
$\text{Max } Z' = \underline{C}\ \underline{X}$	$\text{min } Z = \underline{d}'\ \underline{u}$
$A\ \underline{X} \leq d$	$A'\ \underline{u} \geq \underline{C}'$
$\underline{X} \geq 0$	$\underline{u} \geq 0$

y

$$\underline{C}\ \underline{X}^{o} = \underline{d}'\ \underline{u}^{o} \qquad (7)$$

Introduciendo holguras tanto en el primal como en el dual se puede obtener la siguiente igualdad:

$$A\ \underline{X} + Im\ \underline{X}_s = \underline{d} \qquad (8) \qquad\qquad A'\underline{u} - Im\ \underline{u}_s = \underline{C}' \qquad (10)$$

$$\underline{X},\ \underline{X}_s \geq 0 \qquad (9) \qquad\qquad \underline{u},\ \underline{u}_s \geq 0 \qquad (11)$$

multiplicando (2) por $\underline{u}'$ y (4) por $\underline{X}'$:

$$\underline{u}'\ A\ X + \underline{u}'\ Im\ \underline{X}_s = \underline{u}'\ d;\ \underline{X}'\ A'u - \underline{X}'\ Im\ \underline{u}_s = \underline{X}'\ \underline{C}' \quad (12)$$

$$\underline{u}'\ A\ X + \underline{u}'\ X_s = \underline{d}'\ u \quad ; \quad X'A'u - \underline{X}'\ \underline{u}_s = \underline{C}\ \underline{X}$$

Si existe una solución factible y óptima, se satisface la función objetivo (1) y por lo tanto

$$\underline{u}'\ A\ \underline{X}^o + \underline{u}'^o\ X_s = \underline{X}'^o\ A'\ \underline{u}^o - \underline{X}'^o\ \underline{u}_s^o$$

Las variables de holgura óptimas se obtienen por sustitución en el primal y el dual cuando se conocen $\underline{X}^o$ y $\underline{u}^o$.

De (6) se deduce que

$$\underline{u}^{o\prime}\ X_s^o + \underline{X}^o\ \underline{u}_s^o = 0 \quad (13)$$

lo que implica

$$(\underline{X}^o,\ \underline{u}^o)\begin{pmatrix} \underline{u}_s^o \\ \underline{X}_s^o \end{pmatrix} = 0$$

Cuando esto se cumple se dice X^o y u^o son ortogonales.

1.4 Generación de algoritmos

En conclusión un problema de programación lineal tiene un óptimo si las correspondientes soluciones del primal y dual son factibles básicas y cumplen la propiedad de holgura complementaria, o condición de ortogonalidad.

Se pueden desarrollar ocho diferentes algoritmos en función de lo anterior.

Las ocho combinaciones se pueden representar en una tabla como sigue:

NOMBRE DE LOGARITMO

1	$\bar{P}$	D	O	(F-F) modificado
2	P	$\bar{D}$	O	(F-F) modificado
3	P	D	$\bar{O}$	Algoritmo primal – dual
4	$\bar{P}$	$\bar{D}$	O	(F-F) modificado
5	P	$\bar{D}$	$\bar{O}$	Algoritmo dual simplex
6	$\bar{P}$	D	$\bar{O}$	Algoritmo primal simplex
7	$\bar{P}$	$\bar{D}$	$\bar{O}$	Ford – fulkerson (F–F)
8	P	D	O	Sistema de ecuaciones

donde P es el primal, D el dual y O la ortogonalidad. Si estan testadas se indica la factibilidad que representa el punto de partida del algoritmo ha desarrollarse. La combinación 7 es un generalización de las 1, 2 y 4. En la literatura se pueden encontrar los diferentes algoritmos bajo las bases anteriores.

2. PROGRAMACION DISCRETA

2.1 Objetivo

El problema de programación discreta es determinar $\underline{X}^{o}$, tal que

$$
\begin{aligned}
&\text{maximice } Z = C X \\
&\text{sujeto a } AX = b \\
&\qquad X \geq 0 \\
&\qquad X_s \quad \text{discreto} \\
&\qquad I = \{ i = 1\ 2, \ldots, n \} \\
&\text{SCI}
\end{aligned}
$$

Si S = I el problema es totalmente discreto.

Entre las técnicas de solución se encuentran:

a) Planos de corte

b) Métodos de ramificación progresiva (ramificación y acotación)

c) Búsqueda directa

d) Teoría de grupos

e) Programación dinámica

f) Teoría de dualidad de Balas

La idea de solución al problema de programación discreta, usando los métodos de planos de corte, fue sugerida por Dantzig, Fulkerson y Johnson, consiste en resolver el problema original ignorando los requisitos de integrabilidad. Si la solución que se obtiene es entera, entonces se tiene la solución óptima; en caso contrario, se añaden nuevas restricciones generadas de las anteriores, a esto se le denomina generar cortes, y se procede resolver el problema modificado. Los métodos de planos de corte han presentado grandes problemas de convergencia y por lo tanto computacionales.

Entre los métodos de planos de corte se tiene los correspondientes de Gomory y González-Young, basados en los métodos dual y primal del simplex, respectivamente.

Gomory canalizó la sugerencia y propuso los tres métodos siguientes: Gomory I, II y III. Los Gomory I y II se utilizan cuando todas las variables estan restringidas a tomar valores discretos, exclusivamente.

Cuando el problema de programación discreta considera variables sujetas a valores enteros y continuos, se utiliza el método Gomory III.

González y Young presentaron, simultáneamente, un método basado en la forma primal del simplex, que proporciona, al menos, una solución factible, pero tiene los mismos problemas computacionales que los métodos de Gomory.

Actualmente, el método más utilizado y con más perspectivas y que nos acuparán en este trabajo es el de ramificación y acotación.

Los algoritmos de ramificación y acotación son un nuevo tipo de solución al problema de programación discreta.

Entre los problemas que han sido atacados por esta técnica tenemos: el del agente viajero, la localización de plantas, la asignación de trabajo y la asignación de inversiones y proyectos.

Las condiciones necesarias que debe tener un problema para ser atacado por esta tecnica se establecen en la siguiente sección.

Las técnicas de búsqueda directa, basadas en la búsqueda exhaustiva del óptimo, han sido utilizadas en la solución de problemas pequeños, y su principal desventaja se presenta cuando el tamaño del problema es considerablemente grande.

En la actualidad, la teoría de grupos ha contribuido a la solución de problemas de programación entera, en donde, Gomory ha empezado a publicar los primeros resultados téoricos de los mismos.

La programación dinámica ha sido utilizada en la solución de algunos problemas de programación entera, por ejemplo, el problema de la mochila (Knapsack), en inversiones. Siendo su principal desventaja los requerimientos de memoria y tiempos de computadora, cuando el número de variables de estado es mayor de dos.

Recientemente Egon Balas ha dado a conocer la teoría de dualidad de la programación discreta, basada en la teoría del minimax, la cual establece las mismas propiedades que la teoría de dualidad de la programación lineal. Es de esperar que a partir de esta teoría sea posible generar distintos métodos de solución de los problemas de programación discreta, asemejanza de lo que ocurre en la teoría de dualidad de la programación lineal.

2.2 Fundamentos matemáticos de la técnica de ramificación y acotación

Considérese el siguiente problema de programación discreta:

P : Determinar $\underline{X}^o$ y Z^o tal que

minimice $Z = f(\underline{X})$

sujeto a $\underline{X} \in \Omega_1$

donde

E^n es el espacio n-dimensional

f función real de $\underline{X}$

S_1 conjunto convexo cerrado y acotado con fronteras definidas por hiperplanos en E^n

T_1 conjunto de vectores, finito, no vacío en E^n

$\underline{X}$ vector en E^n

Z valor de la función objetivo

Ω_1 subconjunto finito de S_1 obtenido de la intersección de S_1 y T_1, $\Omega_1 = S_1 \cap T_1$

<u>Defnción</u>. Sea A_j el j-ésimo problema auxiliar continua derivado de P, como sigue:

A_j : Determinar $\underline{X}^*$ (j) y Z^* (j) tal que

minimize $\underline{Z}$ (j) = f ($\underline{X}$)

sujeto a $\underline{X} \epsilon S_j$, j = 1, 2,..., n

Para j = 1, S_1 es conocido y para j > 1, S_j es un subconjunto de S_1, que se define más adelante. Consideremos que existe un algoritmo finito, denominado el subalgoritmo, para la solución del problema A_j. Además, se considera que puede determinarse para el problema P una solución factible, para cada j, por "simple inspección", de la solución del problema A_j. Esta solución se denota pro $\hat{\underline{X}}$ (j), $\hat{Z}$ (j). A la "inspección" se le denomina operación de redondeo.

2.3 Desarrollo del árbol de solución

El algor..t..u de ramificación y acotación es un proceso iterativo que puede interpretarse como un árbol: T (i) = $\left[N (i), A (i)\right]$ donde N (i) es el conjunto de nodos y A (i) es el conjunto de arcos dirigidos al final de la iteración i.

Excepto la primera iteración, se adicionan dos nuevos nodos y arcos dirigidos a los conjuntos N y A.

Asociado con cada nodo j ϵ N (i) está el subconjunto Ω_j de Ω_1 y un subconjunto S_j de S_1, y asociado con cada arco (j, k)ϵA (i) está un conjunto $V_{j,k}$ (por definirse).

Al final de la i-ésima iteración, el subconjunto de N (i), correspondiente a los nodos terminales del árbol, será denotado por C (i). El conjunto C (i) será dividido en tres subconjuntos: F (i), E (i) y R (i) tal que

$$F \cup E \cup R = C$$

donde

F conjunto de nodos factibles o activos

E conjunto de nodos excluidos o no factibles

R conjunto de nodos rechazados

El algoritmo se inicia generando la raíz del árbol (nodo 1), asociándole S_1 y resolviendo A_1. A partir de este y en forma iterativa se adicionan arcos y sus correspondientes nodos, de acuerdo con una operación de ramificación.

Estos arcos dirigidos tienen como origen un nodo convenientemente seleccionado de F (i). Para cada nodo j así creado, se obtiene los valores $\underline{X}^*$ (j), Z* (j) y $\hat{\underline{X}}$ (j), $\hat{Z}$ (j) resolviendo el problema auxiliar A_j y utilizando la operación de redondeo para determinar la solución óptima.

Este procedimiento iterativo termina cuando se obtiene la solución al problema original P, o cuando se tiene suficiente información de que no existe solución. Dicha información está dada por las operaciones de acotación, exclusión y rechazo (por definirse), en unión con las operaciones de ramificación y redondeo.

2.4 Operaciones de ramificaciones y acotación

Definición 1. Operación de ramificación

Sean

$\emptyset$ conjunto vacío,

Ω_j subconjunto no vacío, $\Omega_j \subset \Omega_1$

S_j subconjunto asociado con el nodo j, $S_j \subset S_1$

La operación de ramificación se define como una partición de Ω_j en dos subconjuntos, Ω_r y Ω_{r+1}, tales que

$$\Omega_r \cup \Omega_{r+1} = \Omega_j \qquad (2.1)$$

$$\Omega_r \cap \Omega_{r+1} = \emptyset \qquad (2.2)$$

Esta partición se logra creando dos arcos dirigidos (j, r) y (j, r+1) emanando del nodo j, tal que

$$\Omega_j \cap V_{j,r} = \Omega_r \qquad (2.3)$$

$$\Omega_j \cap V_{j,r+1} = \Omega_{r+1}$$

El siguiente teorema caracteriza $V_{j,r}$ y $V_{j,r+1}$:

Teorema 2.1

Dado Ω_j, las condiciones suficientes de $V_{j,r}$ y $V_{j,r+1}$ para definir una partición que satisface a (2.1) y (2.2) son:

$$V_{j,r} \cap V_{j,r+1} = \Phi \qquad (2.4)$$

$$\Omega_J \subseteq (V_{j,r} \cup V_{j,r+1}) \qquad (2.5)$$

Se asocian a los nodos r y r+1 los subconjuntos S_r y S_{r+1}, definidos como sigue:

$$S_r = S_j \cap V_{j,r} \qquad (2.6)$$

$$S_{r+1} = S_j \cap V_{j,r+1}$$

de donde

$$S_r \cap S_{r+1} = \emptyset$$

Después de que se han desarrollado un número finito de operaciones de ramificación, se espera tener generados nodos t para los cuales Ω_t ha sido reducido a un solo elemento del dominio original Ω_1. Se espera, también que el correspondiente S_t se reduzca al mismo elemento, para que $S_t \equiv \Omega_t$. Se observa que esto es posible para Ω_t puesto que, por hipótesis, Ω_1 es finito; pero no así S_t, que es infinito. De aqui que para garantizar que eventualmente $\Omega_t \equiv S_t$ se han de restringir los conjuntos $V_{j,r}$ y $V_{j,r+1}$ tales que en un número finito de operaciones de ramificación, los nodos con Ω_t conteniendo un solo elemento, esten asociados con un S_t, que contiene el mismo elemento. Finalmente para iniciar y hacer posible la operación de ramificación, los conjuntos Ω_1 y S_1 se asignan la nodo raíz del árbol de soluciones.

El siguiente lema prueba que $\Omega_j \subset S_j$:

Lema 2.1

Sean Ω_r y S_r los conjuntos asociados con r de T (i). Entonces Ω_r es un subconjunto de S_r.

Definición 2. <u>Operación de acotación</u>. Dado el conjunto F (i), la operación de acotación se define por medio de las siguientes acciones:

a) <u>Operación de acotación inferior</u>. Esta consiste de seleccionar el nodo $k \in F(i)$ tal que

$$L_i = Z^*(k) = \min_{j \in F(i)} \left\{ z^*(j) \right\} \qquad (2.8)$$

donde L_i es la cota inferior para el problema P. Se dice que el nodo k es el nodo cota para la iteración i a partir del cual tendrá lugar la siguiente iteración.

b) <u>Operación de acotación superior</u>. Esta consiste en encontrar el valor

$$U_i = \hat{Z}(s) = \min_{j \in F(i)} \left\{ \hat{z}(j) \right\} \qquad (2.9)$$

donde U_i es la cota superior para el problema P.

De lo anterior se tienen los siguientes lemas:

Lema 2.2

Si el nodo j es el predecesor inmediato de r entonces

$\underline{Z}^*(j) \leq Z^*(r)$

Lema 2.3

Sea k el nodo acotado de la iteración i con el valor asociado L_i dado por (2.9). Si $\underline{X}^o$, Z^o es la solución óptima para P, entonces $L_i \leq Z^o$.

Lema 2.4

Si $\hat{\underline{X}}(j)$, $\hat{Z}(j)$ es la solución factible para el problema P, obtenida por una operación de redondeo, en el nodo j, entonces $Z^o \leq \hat{Z}(j)$.

La operación de redondeo da una cota superior a Z^o en cada nodo.

El siguiente teorema resulta de que al final de cada iteración i, la mejor cota inferior correspondiente al nodo k es L_i y la mejor cota superior correspondiente al nodo s es U_i:

Teorema 2.2. En cualquier iteración, $L_i \leq U_i$ y si $L_i = U_i$: se tiene la solución óptima correspondiente a la solución redondeada del nodo s de la iteración i.

Definición 3. Exclusión. La operación de exclusión se define para el nodo terminal r de C (i) para el cual el conjunto correspondiente Ω_r es vacío. Puesto que Ω_r es vacío, no existirá ramificación del nodo r y como parte de la operacion de exclusión el nodo se asigna al conjunto E (i).

Lema 2.5

Si la solución a A_r es no factible, entonces $\Omega_r = \emptyset$.

Definición 4. Rechazo. La operación de rechazo del nodo r asigna a este al conjunto R (i) si se satisface la siguiente condición:

$$Z^* (r) > U_{i-1}$$

Lema 2.6

Si para el nodo r en la iteración i, Z* (r) es más grande que la cota superior de la iteración previa, no es necesaria la ramificación.

Definición 5. Operación de redondeo. Sean $\underline{X}^*$ (j), Z* (j) las soluciones de A_j asociadas con el nodo j. La operación de redondeo obtiene de $\underline{X}^*$ (j), Z* (j) una solución factible $\hat{\underline{X}}$ (j), $\hat{Z}$ (j) para el problema P.

Si la operación es posible para cada nodo j, entonces:

a) La cota superior, U_i, puede ser determinada en cada iteración, haciendo posible la ejecución de la operación de rechazo. Puesto que el nodo rechazado se asigna a R (i) y la selección del nodo por ramificar se hace de F (i), no se requiere ninguna información asociada con el nodo rechazado

b) La determinación de la cota superior, U_i, en cada iteración reduce el intervalo de incertidumbre de la solución óptima Z° puesto que $L_i \leq Z° \leq U_i$

c) El algoritmo da una medida de efectividad para la operación de redondeo. El conjunto C (i) de nodos terminales de la iteración i contiene exactamente i nodos.

Si α es el número de elementos en F (i) y β el número de elementos en R (i), entonces una medida de efectividad para la operación de redondeo es la siguiente:

$$(mde) = \frac{\beta}{\alpha + \beta}$$

donde mde es medida la efectividad.

2.5 Especificación del algoritmo

Se sigue la siguiente secuela:

Paso 1. Hacer i = 1 y crear el nodo j = 1. Resolver A_1. Si la solución es no factible, alto; el problema no tiene solución. Por otro lado si $\underline{X}^*(1) \in \Omega_1$, alto; la solución es óptima. Si $\underline{X}^*(1) \notin \Omega_1$, acotar el nodo 1 con $L_1 = Z^*(1)$. Redondear nodo 1 para obtener $\hat{\underline{X}}(1)$, $\hat{Z}(1)$. Poner $U_1 = \hat{Z}(1)$. Si $L_1 = U_1$, alto; la solución redondeada es óptima. Si, $L_1 < U_1$, asignar nodo 1 a F (1), hacer i = i + 1 e ir al paso i.

Paso i a) Ramificación. Ramificar del nodo acotado j $\in$ F (i). Suprimir nodo j de F (i).

Crear los nodos r y $r + 1$ y los arcos dirigidos (j, r) y $(j, r + 1)$. Resolver los problemas A_r y A_{r+1}, y en ambos casos hacer lo siguiente: Si A_r (A_{r+1}) es no factible, excluir el nodo r $(r + 1)$ asignandolo a $E(i)$. Por otro lado A_r (A_{r+1}) tiene una solución óptima. Si $Z(r)$ $(z^*(r + 1)) > U_{i-1}$, rechazar el nodo $r(r + 1)$ asignándolo a $R(i)$. En caso contrario, $Z^*(r)$ $(Z^*(r + 1)) \leq U_{i-1}$, asignarlo a $F(i)$.

b) Redondeo. Redondear nodo r $(r + 1)$ si fué asignado a $F(i)$ o $R(i)$.

c.1) Cota superior. Sea $U_i = \min\left[U_{i-1}, \hat{Z}(r), \hat{Z}(r + 1)\right]$ para el nodo r $(r + 1) \in E(i)$ o $R(i)$. Rechazar los nodos de $F(i)$ que tienen $Z^*(j) > U_i$ asignándolos a $R(i)$.

c.2) Cota inferior. Seleccionar el nodo $k \in F(i)$ utilizando la operación de acotación inferior. Acotar inferiormente el nodo k con $L_i = Z^*(k)$. Si $L_i = U_i$, alto; la solución factible que da la cota superior es óptima. Si $L_i < U_i$ hacer $i = i + 1$ e ir al paso i.

3. PROBLEMA DE LA INVERSION DE CAPITAL

3.1 Consideraciones generales

Se entiende por problema de inversión de capital aquel en el cual se invierte una cantidad de capital fijo entre un conjunto finito de proyectos, que compiten entre sí por dicho recurso.

En un problema de inversión, se pueden distinguir las siguientes componentes básicas:

1. El conjunto de proyectos entre las cuales va a efectuarse la inversión

2. La erogación total asociada con cada proyecto y su distribución en el tiempo

3. El beneficio total asociado con la aceptación de un proyecto

4. El horizonte de planeación y el número de periodos considerados para la inversión

5. Valor de los presupuestos disponibles para cada periodo.

Los proyectos podrán estar ligados o no, por ciertas formas de dependencia, que pueden ser tecnológicas, operacionales o económicas. Debido a lo anterior, los proyectos se dividen en dependientes e independientes. En este trabajo se estudiaran los proyectos indpendientes.

Son proyectos independientes aquellos en los cuales la erogación y recuperación total de proyectos individuales no se ve afectada por la aceptación o rechazo de otros proyectos del mismo conjunto.

Es posible determinar modelos para los diferentes casos de problemas de inversión.

El tiempo de realización o construcción de un determinado proyecto, afecta también la naturaleza del modelo.

3.2 Casos que se presentan en los problemas de inversión de capital

Los problemas más comunes en inversiones y sus características son:

1. Seleccionar el subconjunto de los n proyectos factibles [illegible] versión, que maximicen el beneficio total si se conocen los desembolsos [illegible]

en m periodos de tiempo y los presupuestos disponibles en cada uno de ellos. Los proyectos seleccionados se empiezan en el primer periodo

2. Es un caso especial del anterior, excepto que se trabaja en un solo periodo

3. Es idéntico al caso 1, pero los proyectos seleccionados pueden deferirse a periodos posteriores.

En este trabajo se presenta el caso 1, considerando los proyectos como independientes.

3.3 Planteamiento matemático del problema

Definiendo

B_j los presupuestos en cada periodo; $j = 1, \ldots, n$

$a_{ij} \geq 0$ capital necesario para el proyecto i en el periodo j; $i = 1, \ldots, m$; $j = 1, \ldots, n$

f_i utilidad asociada con la aceptación del proyecto i o el valor presente neto de la inversión en el proyecto i

y_i variable de decisión: $Y_i = 0, 1$

Se puede plantear el problema en estudio como sigue:

$$P' : \text{Maximizar} \qquad Z = \sum_{i=1}^{m} f_i \; Y_i \qquad (3.1)$$

sujeto a

$$\sum_{i=1}^{m} X_{ij} \quad B_j \qquad (3.2)$$

$$\sum_{j=1}^{n} X_{ij} = Y_i \sum_{j=1}^{n} a_{ij} \tag{3.3}$$

$$0 \leq X_{ij} \leq a_{ij} \quad ; \forall \; (i, j) \in A \tag{3.4}$$

Y_i entero

Si $Y_i = 0$, el proyecto i se rechaza, y si $Y_i = 1$, se acepta.

La ec 3.3 permite expresar P' en términos exclusivamente de X_{ij}, como sigue:

$$P \; : \quad \text{maximizar} \quad Z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_i \quad X_{ij} \tag{3.6}$$

sujeto a

$$\sum_{i=1}^{m} X_{ij} \leq Bj \quad ; \; j = 1, \ldots, n \tag{3.7}$$

$$0 \leq X_{ij} \leq a_{ij}, \forall \; (i, j) \in A \tag{3.8}$$

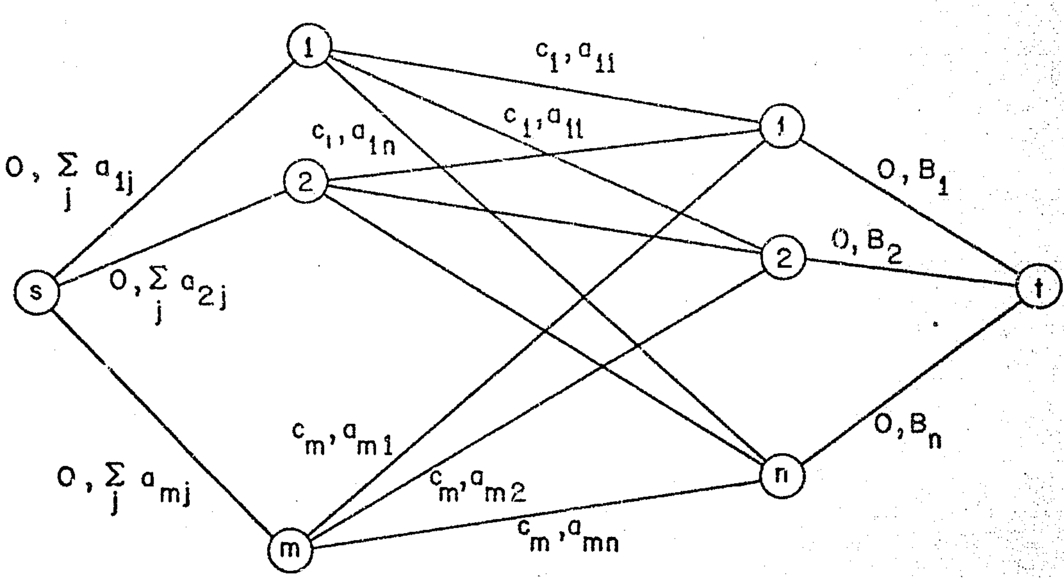
$$Y_i = \frac{\sum_{j=1}^{n} X_{ij}}{\sum_{j=1}^{n} a_{ij}} \; ; \text{ entero} \tag{3.9}$$

donde $C_i = \dfrac{f_i}{\sum_{j=1}^{n} a_{ij}}$ representa la utilidad del proyecto i por unidad de inversión.

Antes de proceder a desarrollar una técnica de solución para el problema P, se demostrará cómo puede representarse gráficamente ut lizando redes.

Si se considerara a los proyectos y periodos como nodos, denominados fuentes y sumidores, respectivamente, y a las parejas ordenadas de los mismos como arcos, cuya capacidad está dada por el costo del proyecto considerado en ese periodo, se puede transformar en una red bipartita G, la cual puede transformarse en una red equivalente con una sola fuente unida a las distintas fuentes por medio de arcos cuya capacidad es $(0, \sum a_{ij})$ y un solo sumidero t unido con arcos a los distintos sumidores cuya capacidad sea (o, B_j).

La red asociada se muestra en la figura siguiente:



Por lo tanto, el problema puede expresarse en términos de la teoría de redes como un problema para determinar el flujo máximo de S a t tal que minimice los costos en la red.

4. APLICACION DE LA PROGRAMACION DISCRETA AL ANALISIS DE INVERSIONES

4.1 Consideraciones generales

Se adoptará en esta sección el algoritmo de ramificación y acotación aplicado al problema de inversiones en proyectos independientes.

Sean los conjuntos S_1, T_1 y Ω_1, definidos como sigue:

$$S_1 = \left[X_{ij} / \sum_{i=1}^{m} X_{ij} \leq B_j, \quad 0 \leq X_{ij} \leq a_{ij} \right] \qquad (4.1)$$

$$T_1 = \left[X_{ij} / \left\{ \sum_{j=1}^{n} X_{ij} / \sum_{j=1}^{n} a_{ij} \right\} = \text{entero}, \right.$$

$$\left. X_{ij} = 0 \quad o \quad a_{ij}, \forall (i, j) \right] \qquad (4.2)$$

$$\Omega_1 = S_1 \cap T_1 \qquad (4.3)$$

El conjunto S_1 es convexo en E^{m+n} obtenido por la intersección de los hiperplanos dados por las ecs 3.7 y 3.8, está acotado, puesto que cada variable X_{ij} de la ec 3.8 está acotada tanto inferior como superiormente.

T_1 es un conjunto no vacío en el mismo espacio de S_1, y es también finito puesto que $X_{ij} = 0$ ó a_{ij}.

Finalmente, de la ec 4.3, Ω_1 es finito, puesto que T_1 es finito.

Por lo tanto, se satisfacen las condiciones del problema general.

Puesto que $a_{ij} \geq 0$ y $B_{ij} \geq 0$, existe al menos una solución factible.

4.2 Operación de ramificación

Dado un cierto nodo l del árbol de solución con los conjuntos asociados Ω_l y S_l, la ramificación se define por su intersección con los conjuntos

$$V_{l\ r} = \left[X_{ij} / X_{kj} = 0, \forall_j \right] \text{ implica } Y_k = 0 \tag{4.4}$$

$$V_{l,r+1} = \left[X_{ij} / X_{kj} = a_{kj}, \forall_j \right] \text{ implica } Y_k = 1 \tag{4.5}$$

para una i = k, dada.

Los conjuntos asi definidos satisfacen las condiciones de suficiencia para formar una partición de Ω_l:

$$V_{l,r} \cap V_{l,r+1} = \left[X_{ij} / X_{kj} = 0, \ X_{kj} = a_{kj}, \forall_j \right] = \emptyset \tag{4.6}$$

$$V_{l,r+1} \cup V_{l,r+1} = \left[X_{ij} / X_{kj} = 0 \text{ ó } a_{kj}, \forall_j \right] \tag{4.7}$$

y puesto que Ω_l es un subconjunto de Ω_1, y también, de 4.1 a 4.3 las variables X_{kj} en Ω_l pueden tomar valores 0 ó a_{kj}.

Así, la intersección de Ω_l con la ec 4.7 es Ω_l y queda satisfecha la segunda condición de suficiencia.

Finalmente, puesto que en cada operación de ramificación n variables X_{ij} son cero o la cota superior, y puesto que el número de variables es finito, eventualmente se obtendrá el nodo terminal t con $S_t \equiv \Omega_l$ conteniendo un solo elemento del dominio de Ω_l, y de aquí todas las soluciones factibles de P pueden ser numeradas en forma similar desarrollando totalmente el árbol de solución.

4.3 El problema auxiliar y el subalgoritmo

En cada iteración del algoritmo de ramificación y acotar asociado con el nodo 1 del árbol de solución, debe resolverse un problema auxiliar conteniendo A_1 derivado de P.

El problema auxiliar A_1 toma la forma:

$$A_1 \quad : \quad \text{maximizar} \qquad Z(1) = \sum_{i=1}^{m} \sum_{j=1}^{n} C_i \; X_{ij} \tag{4.8}$$

sujeto a :

$$\sum_{i=1}^{m} X_{ij} \leq B_{ij} \quad ; \quad j = 1,\ldots, n \tag{4.9}$$

$$0 \leq X_{ij} \leq a_{ij}, \; \forall \, (i, j) \tag{4.10}$$

$$Y_i = \sum_{j=1}^{n} X_{ij} / \sum_{j=1}^{n} a_{ij} = 0 \quad ; \; i \in I_o \tag{4.11}$$

$$Y_i = \sum_{j=1}^{n} X_{ij} / \sum_{j=1}^{n} a_{ij} = 1 \; ; \; i \in I_1 \tag{4.12}$$

donde

$I_o \subseteq N_1$ el subconjunto de nodos de la red G para los cuales $Y_i = 0$ (proyectos rechazados)

$I_1 \subseteq N_1$ subconjunto de N_1 asociado con $Y_i = 1$ (proyecto aceptados)

$\bar{N}_1 = N_1 - I_o - I_1$ subconjunto de proyectos que pueden ser aceptados o rechazados libremente.

Este problema es un programa lineal cuya solución puede obtenerse por inspección. Además, de la ec 4.11, $X^*_{ij} = 0$, $i \in I_o$. De las ecs 4.10 a 4.12, $X^*_{ij} = a_{ij}$, $i \in I_1$.

El problema puede descomponerse en n programas mutuamente independientes, de la forma

$$\text{Maximizar} \qquad Z_j = \sum_{i \in \bar{N}_1} C_i \; X_{ij} \tag{4.13}$$

$$\text{Sujeto a :} \qquad \sum_{i \in \bar{N}_1} X_{ij} \quad \bar{B}_j \tag{4.14}$$

$$0 \leq X_{ij} \leq a_{ij} \tag{4.15}$$

donde $\bar{B}_j = B_j - \sum_{i \in I_1} a_{ij}$

Se considera, sin perder generalidad, que los proyectos con índice $i \in N_1$ están en orden decreciente de su utilidad por unidad de inversión, C_i, tal que : $C_1 \geq C_2 \geq . . . \geq C_m$.

Bajo esta consideración es obvio que la solución óptima al problema dado por las ecuaciones 4.13 a 4.11 puede lograrse simplemente poniendo X_{ij} igual a su cota superior en el orden de los índices i, hasta que el presupuesto B_j se agote. Si $\sum_{i \; N_1} a_{ij} > \bar{B}_j$, entonces una sola variable tomará un valor menor que la cota superior. El problema planteado por las ecuaciones 4.13 a 4.15 puede resolverse para toda j en esta forma.

Así, la solución óptima al problema auxiliar A_1 será $X^*_{ij} = a_{ij}$ si $a_{1j} \leq \bar{B}_j$, o cero, y:

$$
X^*_{rj} = \begin{cases}
0 & \text{si } r \in I_o \\
a_{rj} & \text{si } r \in I_1 \\
a_{rj} \quad \text{si} \quad \sum_{i=1}^{r-1} X^*_{ij} < \bar{B}_j \ \text{y} \ \sum_{i=1}^{r} X^*_{ij} \leq \bar{B}_{ij};\ r \in \bar{N}_1;\ r > 1 \\
\bar{B}_j - \sum_{i=1}^{r-1} X^*_{ij} \quad \text{si} \quad \sum_{i=1}^{r-1} X^*_{ij} < \bar{B}_j \ \text{y} \ \sum_{i=1}^{r} X^*_{ij} > \bar{B}_j;\ r \in \bar{N}_1,\ r > 1 \\
0 \quad \text{si} \quad \sum_{i=1}^{r-1} X_{ij} \geq \bar{B}_j,\ \ r \in \bar{N}_1,\ r > 1
\end{cases} \tag{4.16}
$$

La función objetivo tomará el siguiente valor:

$$
Z^* (1) = \sum_{i \in I_1} f_i + \sum_{i \in \bar{N}_1} \sum_{ij} C_i X^*_{ij} \tag{4.17}
$$

Debe notarse que si para algún problema A_1, su correspondiente $\bar{B}_j$ es menor que cero, A_1 será no factible, y los nodos del árbol deben excluirse.

4.4 Operación de redondeo

Se tiene una solución factible al problema si Y^* es entero; en caso contrario, se puede establecer una solución factible, para P, de la siguiente manera:

$$
Y_i = \begin{cases}
Y^*_i, & \text{si } Y^*_i = 0 \text{ o } 1 \\
0 & \text{si } 0 \quad Y^*_i < 1
\end{cases} \tag{4.18}
$$

El valor de la función objetivo será

$$\hat{Z}(1) = \sum_{i \in I_1} f_i + \sum_{i \in \bar{N}_1} \sum_j C_i \hat{X}_{ij} \tag{4.19}$$

La solución que así se obtiene es factible para P. La operación de redondeo definida por las ecs 4.18 y 4.19 permite rechazar ciertas ramas del árbol.

4.5 Algoritmo de ramificación y acotación

A continuación se presenta el algoritmo .

Paso 1. Hacer $i = 1$, generar nodo 1 resolviendo A_1. Obtener Y_i^*. Si todas las Y_i^* son 1 ó 0, alto; la solución es óptima. Si no acotar nodo 1 con $U_1 = Z^*$. Redondear nodo 1 para obtener $(\hat{Z}, \hat{X}_{ij})$. Hacer $L_1 = \hat{Z}$. Si $L_1 = U_1$, alto; la solución redondeado es óptima. De otra manera $L_1 > U_1$. Hacer $i = i + 1$ y pasar a la etapa i.

Etapa 1. a) Ramificar. Ramificarse del nodo acotado 1. Crear nodo r y r + 1 y los arcos dirigidos (1, r) y (1, r + 1). Resolver el problema A_r con $Y_k = 0$, adicionando k al conjunto I_0 y resolver problema $A_r + {}_1$ con $Y_k = 1$, adicionando k al conjunto I_1. Si $Z^*(r)$ ó $Z^*(r + 1)$ menor que L_{i-1}, rechazar el nodo correspondiente. Si alguno es no factible, excluirlo.

b) Redondeo. Redondear nodo r y r + 1

C_1) Cota inferior. Hacer $L_i = \max\left[L_{i-1}, \hat{Z}(r), \hat{Z}(r + 1)\right]$; rechazar todos los nodos con $Z^* < L_i$

C_2) Cota superior. Seleccionar nodo 1 tal que $Z^* (1) = \max\{Z^* (k)\}$ para los nodos corrientes terminales. Acotar superiormente el nodo 1 con $U_i^k = Z^* (1)$. Si $L_i = U_i$, alto; la solución factible que da la cota inferior es óptima. De otra manera, $L_i < U_i$. Hacer $i = i + 1$ e ir a la etapa i.

La programación y uso de este algoritmo se presenta en el apéndice.

6. REFERENCIAS

1. G. Hadley, "Non-Linear and Dynamic Programing", Addison-Wesley Pubushing Co. Inc., Nueva York (1964)

2. T. C. Hu, "Integer Programming and Network Flows", Addison-Wesley Publishing Co. Inc., Nueva York (1969)

3. F. Ochoa R., "Applications of Discrete Optimization Techniques to Capital Investment and Newtwork Synthesis Problems", M.I.T. (1968)

4. M. Simmonard, "Linear Programming", Prentice Hall Englewood Cliffs, N. J. (1966)

5. G. Hadley, "Linear Programming", Addison-Wesley, Publishing Co. Inc., Nueva York (1969)

6. F. S. Hillier y G. J. Lieberman, "Introduction to Operations Research", Holden-Day, San Francisco (1967)

7. O. R. S. A. "A Branch and Bound Algorithm for Plant Design", Vol 14, (1966), pp 361-368

5. CONCLUSIONES

1. Entre las ventajas que se obtienen al aplicar el algoritmo descrito, cuando el número de proyectos es grande, está la simplificación del cálculo con respecto a otros algoritmos conocidos.

2. Una limitación de esta técnica es la suposición de independencia entre los proyectos, debido a que en la práctica es bastante difícil encontrar proyectos que cumplan esta condición por diversas razones, como son la política de la empresa, los compromisos adquiridos, etc.

3. Con relación a la aplicación del procedimiento de solución, se debe tomar en cuenta que los proyectos deben contener una evaluación completa de los beneficios, y una distribución de los presupuestos y desembolsos por periodo, en valor presente, y ordenados en forma decreciente en función de su relación beneficio-costo.

4. En la práctica, a veces, se vuelve difícil estimar los beneficios, por ser algunas de ellas intangibles, y conocer los presupuestos con exactitud. Así, se debe considerar el problema como aleatorio, porque no se conocen con certeza los financiamientos, disponibilidades de capital, etc.

5. El algoritmo no considera la inversión futura de los excedentes de un periodo.

6. Debido a la lógica del programa de computadora y a las vertricciones de memoria de la máquina para la cual fue diseñado el programa, se dificulta el manejo de más de 100 proyectos.

7. El número de periodos que podría manejar el programa, se puede modificar cuando sea necesario, con solo alterar una instrucción del mis-

mo. Se considera para las empresas privadas un término medio de inversión de 5 años, y las oficinas de gobierno también los hacen por el mismo periodo de tiempo.

8. El programa está diseñado para proporcionar los datos necesarios en la construcción del árbol de solución; obtiene, en forma de lista, los proyectos aceptados y rechazados, las erogaciones requeridas por periodo, y el valor óptimo de la función objetivo, es decir, la utilidad máxima que la empresa obtendría en el caso de que aceptara la combinación de proyectos resultantes (Apéndice).

9. Finalmente, se observa que la dimensión del árbol depende directamente del número de proyectos y no del número de periodos.

START OF SEGMENT

```
      DIMENSION ILA(50),COSTO(20)
      DIMENSION  ZS(50),ZE(50),NY(50),NNA(50).
      DIMENSION  ND(50,10),NV(50,10),Y(50), CT(50)
      DIMENSION  NF(20), NE(5), NR(5), X(50,5)
      DIMENSION  P(5), B(50),  A(50,5). NPR(50)
C     PROGRAMA DE SELECCION OPTIMA DE PROYECTOS CON PRESUPUESTOS CONOCIDOS EN
C     CADA PERIODO
C     LA TECNICA UTILIZADA ES RAMIFICACION Y ACOTACION
C
C     L            NUMERO DE PASO,
C     NF(I)        NUMERO DEL NODO I FACTIBLE
C     NE(I)        NUMERO DEL NODO I EXCLUIDO
C     NR(I)        NUMERO DEL NODO I RECLAZADO
C     BL           LIMITE INFERIOR DE LA SOLUCION EN EL PASO L,
C     BS           LIMITE SUPERIOR DE LA SOLUCION EN EL PASO L,
C
C     NP           NUMERO DE PROYECTOS,
C     NS           NUMERO DE PERIODOS,
C     P(J)         PRESUPUESTO DISPONIBLE EN EL PERIODO J
C     B(I)         BENEFICIO OBTENIDO DEL PROYECTO I
C     A(I,J)       COSTO DEL PROYECTO I EN EL PERIODO J
C
C     N            NUMERO DE NODO
C     ND(N,J)      NUMERO DEL PROYECTO J ACEPTADO O RECHAZADO EN EL NODO N,
C     NV(N,J)      VALOR DE LA DECISION DE ACEPTAR O RECHAZAR EL PROYECTO J EN EL,
C                  NODO N, SI TIENE VALOR 1 SE ACEPTA Y SI TIENE VALOR O SE RECHAZA
C
C     X(I,J]       CANTIDAD ASIGNADA EL PROYECTO I EN EL PERIODO J,
C     CT)I]        COSTO TOTAL DEL PROYECTO I,
C     Y(I)         VALOR DE LA DECISION DE ACEPTACION O RECHAZO DEL PROYECTO I
C     ZS(N)        LIMITE SUPERIOR DEL VALOR DE LA FUNCION EN EL NODO N
C     ZE(N)        LIMITE INFERIOR DEL VALOR DE LA FUNCION EN EL NODO N
C     NPR(N)       NUMERO DEL PROYECTO CON VALOR FRACCIONARIO EN EL NODO N
C     LECTURA DE DATOS
      READ 10, NP, NS, (P(J),J=1,NS)
   10 FORMAT(2I5/(7F10.2))
      PRINT 210, NP,NS, (J,P(J) ,J=1,NS)
  210 FORMAT("  EL NUMERO DE PROYECTOS ES ",I5,"   Y EL NUMERO DE PERIOD
     1OS",I5,/," LOS PRESUPUESTOS SON ",/,(25X,"EN EL PERIODO ",I5,
     2"   DE   ",F10.2))
      READ 11, ((A(I,J),J=1,NS),B(I),I=1,NP)
   11 FORMAT (6F10.2)
      DO 250 I=1,NP
  250 PRINT 211,  I,B(I),(J,A(I,J),J=1,NS)
  211 FORMAT("  EL PROYECTO",I5,"TIENE UN BENEFICIO DE ",F10.2,"  Y UN C
     1OSTO POR PERIODO DE   " ,/, 15X,"PERIODO" 5X,"COSTO"//(15X,I5,5X,
     2 F10.5))
C     EMPIEZA LA PRIMERA ETAPA
      DO 20 J=1,NS
      PA=P(J)
      DO 21 I=1,NP
      IF(PA-A(I,J)) 25,25,26
   25 X(I,J)=PA
      GO TO 20
   26 X(I,J)=A(I,J)
      PA=PA-A(I,J)
   21 CONTINUE
   20 CONTINUE
```

```
      NPR(1)=0
      ZS(1)=0
      ZE(1)=0
      DO 22 I=1,NP
      XA=0
      CT(I)=0
      DO 23 J=1,NS
      XA=XA + X(I,J)
      CT(I)=CT(I) + A(I,J)
   23 CONTINUE
      Y(I)=XA/CT(I)
      IF((1.-Y(I))- 0.001) 27,27,28
   27 ZS(1)=ZS(1)+B(I)
      ZE(1)=ZE(1)+B(I)
      NY(I)=1
      GO TO 22
   28 IF(Y(I)) 29,24,29
   29 NPR(1)=1
      NY(I)=2
      ZS(1)=ZS(1)+B(I)*Y(I)
      GO TO 22
   24 NY(I)=0
   22 CONTINUE
      L=1
      N=1
      NNA(1)=0
      NNF=1
      NL=1
      NF(1)=1
      LFIN=0
      PRINT 32
   32 FORMAT(1H1/////////////"          RESULTADOS DEL PROGRAMA DE SELECCI
     1ON OPTIMA DE PROYECTOS CON PRESUPUESTOS CONOCIDOS EN CADA PERIODO"
     2)
      BL=ZE(1)
      BS=ZS(1)
      IF(NPR(1)) 30,30,31
   30 CALL EXIT
   31 PRINT 111
  111 FORMAT(1H1, 6X," NO. ETAPA",3X,9X," NODOS FACTIBLES",30X," RECHAZA
     1DOS",7X," EXCLUIDOS",2X," L. INFERIOR",3X," L. SUPERIOR"//)
   91 PRINT 95, L, (NF(I),I=1,NNF)
   95 FORMAT(10X,I4,5X," * ",5X,15(I2," ,"))
      IF(NNR) 136,136,137
  137 PRINT 96,(NR(I),I=1,NNR)
   96 FORMAT(1H+,70X,5(I2," ,"))
  136 IF(NNE) 138,138,139
  139 PRINT 140,(NE(I),I=1,NNE)
  140 FORMAT(1H+,92X,2(I2," ,"))
  138 PRINT 141,BL,BS
  141 FORMAT(1H+,68X," *",20X," ,",13X," *",F10.2," *",F10.2)
      IF(LFIN) 700,700,701
  700 DO 97 I=1,5
      NR(I)=0
   97 NE(I)=0
      NNR=0
      NNE=0.
      L=L+1
C     EMPIEZA LA ETAPA I ;CADA ETAPA GENERA DOS NODOS,REVISA SI SON FACTIBLES Y
C     ACTUALIZA TODA LA INFORMACION DEL PROBLEMA
      NPA=0
  100 N=N+1
```

```
      DO 110 I=1,NP
      DO 110 J=1,NS
  110 X(I,J)=0.
      NNA(N)=NNA(NL)+1
      KNL=NNA(NL)
      IF(KNL) 122,122,123
  123 DO 40 J=1,KNL
      ND(N,J)=ND(NL,J)
   40 NV(N,J)=NV(NL,J)
  122 KNA=NNA(N)
      ND(N,KNA)=NPR(NL)
      IF(NPA) 45,45,46
   45 NV(N,KNA)=1
      SE FIJAN A 1 O 0 LAS VARIABLE FRACCIONARIA CON MENOR RELACION BENEFICIO.
      GO TO 48
   46 NV(N,KNA)=0
   48 DO 50 J=1,NS
      PA=P(J)
      DO 51 LL=1,KNA
      KPD=ND(N,LL)
      KPV=NV(N,LL)
      PA=PA-A(KPD,J)*KPV
   51 X(KPD,J)=A(KPD,J)*KPV
      IF(PA) 52,52,53
   52 NNE=NNE+1
      NE(NNE)=N
      GO TO 101
   53 DO 55 I=1,NP
      DO 56 LL=1,KNA
      IF(I-ND(N,LL)) 56,55,56
   56 CONTINUE
      IF(PA-A(I,J)) 58,58,59
   58 X(I,J)=PA
      GO TO 50
   59 X(I,J)=A(I,J)
      PA=PA-A(I,J)
   55 CONTINUE
   50 CONTINUE
      NPR(N)=0
      ZS(N)=0
      ZE(N)=0
      DO 60 I=1,NP
      XA=0
      DO 61 J=1,NS
   61 XA=XA+X(I,J)
      Y(I)=XA/CT(I)
      IF((1-Y(I))- 0.001) 62,62,63
   62 ZS(N)=ZS(N)+B(I)
      ZE(N)=ZE(N)+B(I)
      NY(I)=1
      GO TO 60
   63 IF( Y(I))   64,64,65
   64 NY(I)=0
      GO TO 60
   65 NPR(N)=I
      NY(I)=2
      ZS(N)=ZS(N)+B(I)*Y(I)
   60 CONTINUE
      IF(BL    -ZS(N)) 68,68,69
   69 NNR=NNR+1
      NR(NNR)=N
```

```
      GO TO 101
   68 IF(BL    -ZE(N)) 70,71,71
   70 BL=ZE(N)
   71 NNF=NNF+1
      NF(NNF)=N
  101 IF(NPA) 75,75,76
   75 NPA=1
      GO TO 100
   76 DO 77 I=1,NNF
      IM=I
      IF(NF(I) -NL) 77,79,77
   77 CONTINUE
   79 NNF=NNF-1
      DO 80 J=IM,NNF
   80 NF(J)=NF(J+1)
      NF(NNF+1)=0
      DO 135  I=1,NNF
      IP = I
      MP=NF(I)
      IF(BL-ZS(MP))  135,135,132
  135 CONTINUE
      GO TO 151
  132 NNF=NNF-1
      NNR=NNR+1
      NR(NNR)=MP
      DO 133 K=IP,NNF
  133 NF(K)=NF(K+1)
      NF(NNF+1)=0
  151 BS=0.
      DO 85 I=1,NNF
      MP=NF(I)
      IF(BS    -ZS(MP)) 86,86,85
   86 BS=ZS(MP)
      NL=MP
   85 CONTINUE
      IF((BS    -BL    )- 0.001)  90,90,91
   90 LFIN=1
      GO TO 91
  701 PRINT 99,BL
   99 FORMAT(1H1///////"   EL VALOR OPTIMO DE LA FUNCION OBJETIVO ES",F2
     10.5)
      PRINT 128
  128 FORMAT( 1H1,/// " RESULTADOS PARA FORMAR EL ARBOL DE RAMIFICACION
     1Y ACOTACION"//" EL NUMERO DE PROYECTOS FIJOS EN CADA NODO ESTA DAD
     2O EN LA LISTA DE N. DE PROYECTO"// " LOS PROYECTOS FIJOS Y SUS VAL
     3ORES CORRESPONDIENTES ESTAN DADOS EN LAS LISTAS "/ " DE N  V , EN
     4DONDE LOS VALORES PUEDEN SER 0 O 1 PARA LOS PROYECTOS" / " ACEPTAD
     5OS O RECHAZADOS RESPECTIVAMENTE" /// "        NO. NODO   L. INFERIOR
     6 L. SUPERIOR PROYECTO  N    V    N    V    N    V    N    V    N    V
     7   N    V    N    V    N    V    N    V")
                                                                 SEGMENT:
      DO 129  I=1,N
      K=NNA(I)
      PRINT 127, I,ZE(I),ZS(I),K,(ND(I,J),NV(I,J),J=1,K)
  127 FORMAT(8X,I3,4X,F10.3,4X,F10.3,4X,I3,5X,9(I2,2X,I2,3X))
  129 CONTINUE
C     RELOCALIZACION DE LOS NODOS OPTIMOS
C
      PRINT 1050
 1050 FORMAT(1H1////////////////50X,"LOS NODOS OPTIMOS SON")
      DO 1003 I=1,N
      IF(ZE(I)=ZS(I).AND.ZS(I)=BS)  GO TO 1002
```

```
      GO TO 1001
 1002 PRINT 1051, I,ZE(I)
 1051 FORMAT(1H1,"  EL NODO ",I5,"  CON UN VALOR DE LA FUNCION OBJETIVO
     1DE ",F10.5/"  Y CON LA SIGUIENTE LISTA DE PROYECTOS ACEPTADOS Y RE
     2CHAZADOS"///"                    SI EL VALOR ES  1  SE ACEPTA "/"
     3          SI ES  0  SE RECHAZA"///)
C      DETERMINACION DE LOS VALORES DE ACEPTACION O RECHAZO  DE CADA UNO DE LOS
C                          PROYECTOS EN ESTE NODO OPTIMO
                                                                     SEGMENT
      DO 2001 MU=1,NP
 2001 ILA(MU)=0
      K=NNA(I)
      DO 1003 J=1,NS
      PA=P(J)
      DO 1004 L=1,K
      IF(NV(I,L)=1) GO TO 1005
      GO TO 1004
 1005 LU=ND(I,L)
      PA=PA-A(LU,J)
      ILA(LU)=1
 1004 CONTINUE
      DO 1010 LU2=1,NP
      LU4=0
      DO 1011 LU3=1,K
      IF(LU2=ND(I,LU3)) LU4=1
 1011 CONTINUE
      IF(LU4)1010,1021,1010
 1021 IF(PA)1022,1023,1022
 1022 PA=PA-A(LU2,J)
      ILA(LU2)=1
 1010 CONTINUE
 1003 CONTINUE
 1023 PRINT 1052,(IM,ILA(IM),IM=1,NP)
 1052 FORMAT("      PROYECTO       VALOR "/(5X,I5,I11))
C      VALUACION DE LA EROGACION REQUERIDA POR ESTA ALTERNATIVA
      DO 4102 J33=1,NS
 4102 COSTO(J33)=0
      DO 4101 IM=1,NP
      IF(ILA(IM)=0) GO TO 4101
      DO 4103 J33=1,NS
 4103 COSTO(J33)=COSTO(J33)+A(IM,J33)
 4101 CONTINUE
      PRINT 4105, (J33,COSTO(J33),J33=1,NS)
 4105 FORMAT(///"    LAS EROGACIONES POR PERIODO REQUERIDAS POR ESTA ALT
     1ERNATIVA SON "//10X,"PERIODO","        EROGACION"//(I14,6X,F12.2))
 1001 CONTINUE
      CALL EXIT
      END
                                                                     SEGMENT
```

EL NCDO 35 CCN UN VALCR DE LA FUNCICN OBJETIVC DE 70.00000
Y CCN LA SIGUIENTE LISTA DE PRCYECTCS ACEPTADCS Y RECHAZADCS

SI EL VALCR ES 1 SE ACEPTA
SI ES 0 SE RECHAZA

PRCYECTC	VALCR
1	1
2	1
3	0
4	1
5	1
6	0
7	1
8	0
9	0
10	0

LAS ERCGACICNES PCR PERICDC REQUERIDAS PCR ESTA ALTERNATIVA SCN

PERICDC	ERCGACICN
1	48.00
2	20.00

EL VALCR CPTIMC DE LA FUNCICN CBJETIVC ES 70.00000

NO. ETAPA	NODOS FACTIBLES
1	* 1 *
2	* 3 *
3	* 4 * 5 *
4	* 5 , 6 , 7 ,
5	* [illegible]
6	* 7 , 9 , 11 ,
7	* 8 , 9 , 11 ,
8	* 8 , 9 ,
9	* 6 , 17 ,
10	* 8 , 19 ,
11	* 6 , 21 ,
12	* 21 , 23 ,
13	* 21 , 24 ,
14	* 23 , 26 ,
15	* 27 ,
16	* 31 ,
17	* 33 ,
18	* 35 ,

RECHAZADOS	EXCLUIDOS	L. INFERIOR	L. SUPERIOR
*	,	* 47.00 *	77.62
*	, 2 ,	* 47.00 *	77.20
*	,	* 59.00 *	74.47
*	,	* 59.00 *	73.85
* 7 ,	,	* 70.00 *	73.10
*	, 10 ,	* 70.00 *	72.97
* 12 ,	,	* 70.00 *	72.79
* 14 , 15 ,	,	* 70.00 *	72.40
*	, 16 ,	* 70.00 *	72.38
*	, 18 ,	* 70.00 *	72.36
*	, 20 ,	* 70.00 *	72.31
* 22 ,	,	* 70.00 *	71.85
* 25 ,	,	* 70.00 *	71.58
* 27 ,	,	* 70.00 *	71.13
* 29 ,	, 28 ,	* 70.00 *	70.58
*	, 30 ,	* 70.00 *	70.54
*	, 32 ,	* 70.00 *	70.51
*	, 34 ,	* 70.00 *	70.00

```
EL NUMERO DE PROYECTOS ES     10     Y EL NUMERO DE PERIODOS      2
LOS PRESUPUESTOS SON
                              EN EL PERIODO       1    DE        50.00
                              EN EL PERIODO       2    DE        20.00
EL PROYECTO      1TIENE UN BENEFICIO DE        15.00  Y UN COSTO POR PERIODO DE
              PERIODO       COSTO

                  1          6.00000
                  2          2.00000
EL PROYECTO      2TIENE UN BENEFICIO DE        17.00  Y UN COSTO POR PERIODO DE
              PERIODO       COSTO

                  1          6.00000
                  2          6.00000
EL PROYECTO      3TIENE UN BENEFICIO DE        15.00  Y UN COSTO POR PERIODO DE
              PERIODO       COSTO

                  1          6.00000
                  2          7.00000
EL PROYECTO      4TIENE UN BENEFICIO DE        12.00  Y UN COSTO POR PERIODO DE
              PERIODO       COSTO

                  1          6.00000
                  2          6.00000
EL PROYECTO      5TIENE UN BENEFICIO DE        14.00  Y UN COSTO POR PERIODO DE
              PERIODO       COSTO

                  1         12.00000
                  2          3.00000
EL PROYECTO      6TIENE UN BENEFICIO DE        40.00  Y UN COSTO POR PERIODO DE
              PERIODO       COSTO

                  1         30.00000
                  2         35.00000
EL PROYECTO      7TIENE UN BENEFICIO DE        12.00  Y UN COSTO POR PERIODO DE
              PERIODO       COSTO

                  1         18.00000
                  2          3.00000
EL PROYECTO      8TIENE UN BENEFICIO DE        17.00  Y UN COSTO POR PERIODO DE
              PERIODO       COSTO

                  1         54.00000
                  2          7.00000
EL PROYECTO      9TIENE UN BENEFICIO DE        14.00  Y UN COSTO POR PERIODO DE
              PERIODO       COSTO

                  1         48.00000
                  2          4.00000
EL PROYECTO     10TIENE UN BENEFICIO DE        10.00  Y UN COSTO POR PERIODO DE
              PERIODO       COSTO

                  1         36.00000
                  2          3.00000
```

RESUL

EL NU

LOS F
DE N
ACEPT

1 1

1 C

RESULTADOS PARA FORMAR EL ARBOL DE RAMIFICACION Y ACOTACION

EL NUMERO DE PROYECTOS FIJOS EN CADA NODO ESTA DADO EN LA LISTA DE N. DE PROYECTO

LOS PROYECTOS FIJOS Y SUS VALORES CORRESPONDIENTES ESTAN DADOS EN LAS LISTAS
DE N Y V. EN DONDE LOS VALORES PUEDEN SER 0 O 1 PARA LOS PROYECTOS
ACEPTADOS O RECHAZADOS RESPECTIVAMENTE

NO. NODO	L. INFERIOR	L. SUPERIOR	PROYECTO	N	V	N	V	N	V	N	V
1	47.000	77.515	1	6	0						
2	0.000	0.000	1	6	1						
3	47.000	77.200	1	6	0						
4	59.000	79.467	2	6	0	7	1				
5	47.000	73.102	2	6	0	7	0				
6	58.000	73.846	3	6	0	7	1	5	1		
7	59.000	67.230	3	6	0	7	1	5	0		
8	70.000	72.308	4	6	0	7	1	5	1	4	1
9	56.000	72.404	4	6	0	7	1	5	1	4	0
10	0.000	0.000	3	6	0	7	0	8	1		
11	47.000	72.769	3	6	0	7	0	8	0		
12	14.000	39.077	4	6	0	7	0	8	0	5	1
13	47.000	72.790	4	6	0	7	0	8	0	5	0
14	42.000	54.385	5	6	0	7	0	8	0	5	0
15	47.000	69.200	5	6	0	7	0	8	0	4	0
16	0.000	0.000	5	6	0	7	1	5	1	4	0
17	58.000	72.385	5	6	0	7	1	5	1	4	0
18	0.000	0.000	6	6	0	7	1	5	1	4	0
19	58.000	72.359	6	6	0	7	1	5	1	4	0
20	0.000	0.000	7	6	0	7	1	5	1	4	0
21	58.000	71.846	7	6	0	7	1	5	1	4	0
22	53.000	68.958	5	6	0	7	1	5	1	4	1
23	70.000	70.557	5	6	0	7	1	5	1	4	1
24	56.000	71.583	8	6	0	7	1	5	1	4	0
25	58.000	58.000	8	6	0	7	1	5	1	4	0
26	58.000	71.125	9	6	0	7	1	5	1	4	0
27	56.000	56.000	9	6	0	7	1	5	1	4	0
28	0.000	0.000	10	6	0	7	1	5	1	4	0
1 1											
29	58.000	58.000	10	6	0	7	1	5	1	4	0
1 0											
30	0.000	0.000	6	6	0	7	1	5	1	4	1
31	70.000	70.538	6	6	0	7	1	5	1	4	1
32	0.000	0.000	7	6	0	7	1	5	1	4	1
33	70.000	70.515	7	6	0	7	1	5	1	4	1
34	0.000	0.000	8	6	0	7	1	5	1	4	1
35	70.000	70.000	8	6	0	7	1	5	1	4	1

E N. DE PROYECTO

LAS LISTAS
S

N	V	N	V	N	V	N	V	N	V	N	V	N	V	
5	1													
5	0													
5	1	4	1											
5	1	4	0											
8	1													
8	0													
8	0	9	1											
8	0	9	0											
8	0	9	0	10	1									
8	0	9	0	10	0									
5	1	4	0	8	1									
5	1	4	0	8	0									
5	1	4	0	8	0	9	1							
5	1	4	0	8	0	9	0							
5	1	4	0	8	0	9	0	10	1					
5	1	4	0	8	0	9	0	10	0					
5	1	4	1	3	1									
5	1	4	1	3	0									
5	1	4	0	8	0	9	0	10	0	3	1			
5	1	4	0	8	0	9	0	10	0	3	0			
5	1	4	0	8	0	9	0	10	0	3	1	2	1	
5	1	4	0	8	0	9	0	10	0	3	1	2	0	
5	1	4	0	8	0	9	0	10	0	3	1	2	1	
5	1	4	0	8	0	9	0	10	0	3	1	2	1	*
5	1	4	1	3	0	8	1							
5	1	4	1	3	0	8	0							
5	1	4	1	3	0	8	0	9	1					
5	1	4	1	3	0	8	0	9	0					
5	1	4	1	3	0	8	0	9	0	10	1			
5	1	4	1	3	0	8		9	0	10	0			

ARBOL DE SOLUCION