17
Tej

Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

FALLA DE ORIGEN

UN SISTEMA INTERACTIVO PARA EL PROBLEMA DE PROGRAMACION ENTERA UTILIZANDO ENUMERACION IMPLICITA.

T E S I S
Que para obtener el Titulo de
A C T U A R I O
p r e s e n t a

RICARDO CRESPO MENDOZA

México, D. F. 1991

INDICE

PROLOGO

El objetivo de la presente tesis es el desarrollar un sistema interactivo para el problema de programación entera puro utilizando la enumeración implícita, es decir, que trataremos con el Problema de Programación Entera Lineal (P.P.L.E.) con Variables 0-1.

El trabajo escrito consta de tres capitulos. El primero de ellos tiene la finalidad de mostrar la importancia de la programación lineal entera. En el segundo capitulo se desarrollan el concepto de enumeración implícita, el esquema de enumeración y once pruebas sondeadoras. El tercer capitulo da una descripción del programa.

El sistema provee al usuario de un esquema de enumeración el cual asegura que todas las combinaciones binarias al P.P.L.E. sean enumeradas (explicita o implicitamente) así como de once pruebas sondeadoras diseñadas para hacer que la enumeración sea razonablemente implicita.

Se ha diseñado el sistema de manera que éste sea amigable al usuario, de tal forma que la introducción del problema así como la edición de éste se efectúe de manera sencilla. El sistema valida cada dato y en caso de que el usuario introduzca datos incompatibles, muestra la(s) posible(s) causa(s) del error y permite volver a introducir el dato requerido.

El sistema brinda la opción de resolver el P.P.L.E. de manera automática, ie, sin necesidad de que el usuario intervenga en la búsqueda de la(s) solución(es) óptima(s) al P.P.L.E., así como dos opciones de ayuda en las cuales el usuario podrá consultar información acerca de cómo se realiza cada prueba y consultar sugerencias y estrategias para la resolución del P.P.L.E.

Además, al efectuar cada prueba sondeadora el sistema proporcionará información con la cual el usuario podrá hacer una mejor selección de la siguiente prueba a aplicar e información con la cual se podrá construir fácilmente el árbol que describe el esquema de enumeración.

Es de esperar que esta tesis sea un buen apoyo didáctico para el alumno que estudie la programación lineal entera con un enfoque de enumeración implícita. Para las personas que conocen el tema, el sistema podría brindar un apoyo en el desarrollo de estrategias en la resolución del P.P.L.E. dado que brinda gran flexibilidad en la secuencia de aplicación de las pruebas sondeadoras.

1 OPTIMIZACION ENTERA

1.1 INTRODUCCION

Cualquier problema de toma de decisiones, con un objetivo que deba ser maximizado ó minimizado, en el cual las variables de decisión (las cuales deben ser cuantificables) deban tomar valores discretos (no fraccionales) puede ser clasificado como un problema de optimización entera.

Un problema de optimización entera es clasificado como lineal si, al relajar las condiciones de integridad de las variables, el problema resultante es estrictamente lineal.

La importancia de la optimización entera en cuanto a la resolución de problemas prácticos surge como consecuencia de los grandes avances en el campo de la investigación de operaciones, en particular de la programación lineal. Fue entonces cuando se reconoció la necesidad de resolver problemas de programación en los cuales algunas o todas la variables de decisión debieran tomar valores enteros. En la práctica se dan problemas en los cuales tanto las actividades como los recursos son indivisibles, este es el caso de máquinas, barcos, camiones, etc. Además, muchos problemas requieren la determinación de decisiones de "sí o no", las cuales pueden ser consideradas como los valores 0-1 de una variable entera.

Fue en 1958 que Gomory desarrolló la primera técnica de programación entera finita para resolver problemas enteros lineales y desde entonces se han desarrollado una gran cantidad de algorítmos para la programación entera.

1.2 DEFINICION MATEMATICA DEL PROBLEMA DE PROGRAMACION ENTERA.

El problema general de programación entera puede se definido como:

Maximizar (ó Minimizar)

$$Z = g_0(x_1,\ldots,x_n)$$

s.c

$$g_i(x_1,\ldots,x_n) \le b_i, \quad i \in M = \{1,2,\ldots,m\}$$

$$x_j \ge 0, \quad j \in N = \{1,2,\ldots,n\}$$

$$x_j \text{ entera }, \quad j \in I \subseteq N$$

Si I = N, es decir, todas las variables x_j deben tomar valores enteros, el problema es llamado **problema de programación entera puro.** De otra forma, es decir, si I es subconjunto propio de N, entonces se está tratando con un **problema de programación entera mixto.**

Los desarrollos más importantes en el campo de la programación entera se han realizado en los casos en donde las funciones g_i, con i en {0} U M, son lineales. Podemos escribir el problema general de programación lineal entera de la siguiente forma:

Maximizar (ó Minimizar)

$$Z = \sum_{j \in N} c_j x_j$$

s.c

$$\sum_{j \in N} a_{ij} x_j \le b_i, \quad i \in M$$

$$x_j \ge 0, \quad j \in N$$

$$x_j \text{ entera }, \quad j \in I \subseteq N$$

Tanto la clase de problemas puros como la clase de problemas mixtos se dividen a su vez en dos subclases:

a) Los P.P.E. en los cuales todas las variables enteras de decisión están restringidas a tomar valores 0 ó 1 (estos problemas se conocen como problemas 0-1).

b) Los P.P.E. en los cuales las variables enteras de decisión están restringidas a tomar valores no-negativos.

En la presente tesis trabajaremos con el Problema de Programación Entera Lineal Puro con Variables 0-1.

Para ilustrar la versatilidad de los problemas de programación entera se presentan a continuación algunos ejemplos:

1.3 EJEMPLOS

1) Variables con Valores Discretos.

Consideremos un modelo de programación lineal con el requerimiento adicional de que algunas de las variables puedan tomar sólo ciertos valores discretos.

En general, si x_i es una variable de decisión del modelo que está restringida a tomar valores en el conjunto $\{a_1,\ldots,a_k\}$ la restricción de que x_i pertenezca al conjunto $\{a_1,\ldots,a_k\}$ es equivalente a:

$$x_i - (a_1 y_1 + \ldots + a_k y_k) = 0$$

$$y_1 + \ldots + y_k = 1$$

$$y_j = 0 \text{ ó } 1 \text{ para cada } j$$

Se pueden aumentar tales restricciones a las ya existentes para cada variable restringida a tomar valores discretos. Esto transforma el problema en un problema entero.

2) Problemas de Surtido Mínimo.

En adición a las restricciones usuales en un modelo de programación lineal, supongamos que existen restricciones de la forma: la variable x_j debe tomar valores 0 ó si toma un valor positivo éste debe ser mayor o igual que alguna cota específica l_j. Restricciones de este tipo surgen cuando el modelo incluye variables que representan los montos de algunas materias primas utilizadas, y los distribuidores de estas materias primas proveerán únicamente cantidades mayores o iguales a esas cotas mínimas especificadas.

Sea M_j un número positivo muy grande o alguna cota superior para el valor de x_j en la solución óptima del problema. La restricción de que x_j es 0 ó mayor o igual a l_j es equivalente a:

$$x_j - l_j y_j \geq 0$$

$$x_j - M_j y_j \leq 0$$

$$y_j = 0 \text{ ó } 1$$

Restricciones como ésta pueden ser introducidas en el modelo para cada variable con estas características. Esto transforma el modelo en un modelo de programación entera.

3) **Problemas de Cargo Fijo.**

En los modelos de programación lineal las variables de decisión normalmente representan los niveles a los cuales varias actividades son llevadas a cabo. Se supone que la función objetivo a ser minimizada (ó maximizada) es una función lineal de las variables.

En muchos problemas prácticos, el costo de ejecutar una actividad como una función del nivel a la cual ésta es ejecutada puede tener la forma: el costo de ejectutar la actividad, al nivel x_j es 0 si $x_j = 0$ y $f_j + c_j x_j$ si $x_j > 0$. f_j es un cargo fijo en el cual se incurre sólo cuando la j-ésima actividad es ejecutada a un nivel positivo. Suponemos que f_j es no-negativa para toda j. Sean las restricciones de las variables de decisión

$$Ax = b$$

$$x \geq 0$$

El problema de encontrar una solución que minimice el costo total (el costo variable más los cargos fijos de las variables con valor positivo en la solución) es conocido como el **problema de programación lineal de cargo fijo.** Las f_j son conocidas como **cargos fijos.** Las c_j son los **coeficientes de costo de las variables.** La función objetivo es:

$$Z(x) = \sum_{j=1}^{n} c_j x_j + \sum_{j \in J(x)} f_j$$

en donde para la solución entera **x**, J (**x**) = {j : x_j > 0}. Claramente Z (**x**) no es continua en el origen. Debido a la complejidad (no linealidad) de la función objetivo, estos problemas no pueden ser resueltos por el método simplex.

Los problemas de cargo fijo pueden ser formulados como problemas enteros. Sea M un número positivo muy grande o una cota superior para los valores de las variables en la solución óptima del problema. Consideremos el sistema de restricciones:

$$x_j - M_j y_j \leq 0$$

$$x_j \geq 0, y_j = 0 \text{ ó } 1$$

Claramente, cuando y_j es igual a 0, x_j debe ser igual a 0 bajo estas restricciones. Alternativamente y_j es forzada a tomar valor 1 cuando $x_j > 0$. Esta observación y el hecho de que los f_j son todos no-negativos implica que el problema de cargo fijo es equivalente al siguiente problema de programación lineal entera mixto:

$$\text{Minimizar } \sum c_j x_j + \sum f_j y_j$$

$$\text{s.c}$$

$$Ax = b$$

$$x_j - M y_j \leq 0 \text{ para toda } j$$

$$x_j \geq 0 \text{ para toda } j$$

$$y_j = 0 \text{ ó } 1 \text{ para toda } j$$

4) Problemas de Localización de Plantas.

Los problemas de localización de plantas forman una clase importante de problemas prácticos que pueden ser formulados como problemas enteros. El problema más simple de esta clase tiene la siguiente estructura: existen n sitios en una región que requieren un producto. La demanda para el producto en el sitio i es d_i unidades, $i = 1,\ldots,n$. La demanda tiene que ser satisfecha manufacturando el producto dentro de la región. Para satisfacer la demanda se necesita establecer m ó menos plantas para manufacturar el producto, donde m está especificado. El costo por construir una planta manufacturera en el sitio i es f_i pesos. Si una planta es construída en el sitio i, k_i unidades es su capacidad de producción. El cargo fijo por establecer una ruta para transportar el producto del sitio i al sitio j es f_{ij} pesos. Si una ruta es establecida para transportar el producto del sitio i al j, k_{ij} unidades es la capacidad de la ruta, y c_{ij} pesos es el costo por transportar una unidad del producto de i hacia j. En la práctica m será mucho menor que n, y el producto será mandado de los sitios donde éste es manufacturado hacia todos los otros sitios en la región. El problema es determinar un subconjunto óptimo de sitios para localizar las plantas y el plan de transporte de manera que se minimice el costo total de construir las plantas, establecer las rutas y transportar el producto.

El problema puede ser formulado como un problema entero utilizando variables de decisión con las siguientes interpretaciones:

$y_i = 1$ si una planta es localizada en el sitio i

$= 0$ en otro caso

$y_{ij} = 1$ si se establece una ruta de transporte del sitio i al sitio j

$= 0$ en otro caso

$x_{ij} =$ cantidad de producto transportado del sitio i al sitio j

Entonces, la formulación del problema es:

$$\text{Minimizar } \sum_i \sum_j c_{ij} x_{ij} + \sum_i f_i y_i + \sum_i \sum_j f_{ij} y_{ij}$$

s.c

$$\sum_j x_{ij} - k_i y_i \leq 0 \quad \text{para toda } i$$

$$x_{ij} - k_{ij} y_{ij} \leq 0 \quad \text{para toda } i, j$$

$$\sum_i x_{ij} \geq d_j \quad \text{para toda } j$$

$$\sum_i y_i \leq m$$

$$x_{ij} \geq 0 \quad \text{para toda } i, j; \quad y_i, y_{ij} = 0 \text{ ó } 1$$

5) Problemas de Adquisición.

Esta clase de problemas tiene la siguiente estructura: existen m tipos de equipos disponibles. Hay almacenados a_i unidades del tipo i, i = 1,2,...,m. Un presupuesto de R pesos está disponible para comprar cantidades de equipo extra. Una unidad del tipo i cuesta p_i pesos. El equipo puede ser comprado y distribuído en cantidades enteras. Existen n estaciones que utilizan estos equipos. La j-ésima estación puede usar un máximo de b_j unidades de todos los tipos de equipos (considerados conjuntamente). La ganancia resultante de asignar una unidad de tipo i a la j-ésima estación es c_{ij} pesos. El problema es determinar cuántas unidades de cada tipo de equipo deben ser compradas y cómo distribuir estos equipos entre las estaciones.

Definamos las variables de la siguiente forma:

y_i - número de unidades de equipo tipo i a ser compradas

x_{ij} - número de unidades de equipo tipo i distribuídas en la estación j

Entonces del problema es:

$$\text{Maximizar} \quad \sum_i \sum_j c_{ij} x_{ij}$$

s.c

$$\sum_j x_{ij} - y_i \le a_i; \quad i = 1, \ldots, m$$

$$\sum_i x_{ij} \le b_j; \quad j = 1, \ldots, n$$

$$\sum_i p_i y_i \le R$$

$$x_{ij} \ge 0, \quad y_i \ge 0; \quad \text{todas las variables enteras}$$

6) Restricciones de "Una o la Otra".

En el ejemplo 2), se vió una restricción en la cual una de las condiciones: $x_j = 0$ ó $x_j >= l_j$ tenia que cumplirse. Tales condiciones son conocidas como **restricciones de una o la otra.**

Una generalización de estas restricciones requiere que k restricciones dentro de un conjunto dado de r restricciones tengan que cumplirse, donde k es un número conocido. Supongamos que las restricciones son $g_1(\mathbf{x}) >= 0, \ldots, g_r(\mathbf{x}) >= 0$, donde $g_1, \ldots, g_r$ son funciones dadas de las variables de decisión $\mathbf{x}$. Sea L_i un número positivo muy grande tal que $-L_j$ es una cota inferior para g_j (**x**) sobre el conjunto de soluciones factibles. El requerimiento de que k de estas restricciones tengan que ser satisfechas es equivalente a:

$$g_1(x) + L_1 y_1 \geq 0$$

$$\vdots$$

$$g_r(x) + L_r y_r \geq 0$$

$$y_1 + \dots y_r = r - k$$

$$y_j = 0 \text{ ó } 1 \text{ para cada } j$$

Así, las restricciones de este tipo pueden ser formuladas utilizando variables 0-1.

7) Problemas de Selección de Proyectos.

Este modelo trata con situaciones en las cuales hay varios proyectos esperando aprobación y el problema es determinar de ellos un subconjunto óptimo para ser realmente aprobado. El horizonte de planeación es n periodos. Si el i-ésimo proyecto es aprobado éste requiere un presupuesto de c_{it} pesos en el t-ésimo periodo, t = 1,...,n. El total de fondos disponible en el t-ésimo periodo son c_t pesos. La ganancia total en el caso de que el i-ésimo proyecto sea aprobado es p_i pesos.

Definamos las variables enteras con las siguientes interpretaciones:

$$x_i = 1 \text{ si el proyecto } i \text{ es aprobado}$$
$$= 0 \text{ en otro caso}$$

Entonces el problema es:

$$\text{Maximizar } \sum p_i x_i$$

$$\text{s.c}$$

$$\sum_i c_{it} x_i \leq c_t; \quad t = 1, \ldots, n$$

$$x_i = 0 \text{ ó } 1 \quad \text{para toda } i$$

Restricciones adicionales sobre el uso de otros recursos durante cada periodo pueden ser añadidas al modelo.

8) Problemas de Elección Múltiple.

Consideremos el modelo usual de programación lineal en el cual hay varias actividades posibles. Sea x_j el nivel al cual la j-ésima actividad es realizada. Las variables x_j tienen que ser no-negativas y satisfacer algunas restricciones lineales. Además supongamos que las actividades están agrupadas y que exactamente una actividad de cada grupo debe realizarse.

Sea M un número positivo muy grande o una cota superior para el nivel de cada actividad. Supongamos que $J = (1,\ldots,k)$ forma un conjunto de actividades. Definamos:

$$y_j = 1 \quad \text{si la } j\text{-ésima actividad es realizada}$$

$$= 0 \quad \text{en otro caso}$$

Entonces, la restricción de que exactamente una actividad del grupo J deba ser ejecutada es equivalente a:

$$x_j - M y_j \leq 0 \quad \text{para toda } j \text{ en } J$$

$$\sum_{j \in J} y_j = 1$$

$$x_j \geq 0, \ y_j = 0 \text{ ó } 1; \quad \text{para toda } j \text{ en } J$$

Restricciones similares se deben incluir para cada grupo de actividades de los cuales sólo una actividad tenga que ejecutarse. El modelo final es un problema de programación entera mixta.

9) Problemas de Secuenciación.

Estos problemas surgen al querer determinar una secuencia óptima para trabajos procesados en una máquina. Existen n trabajos. Sea p_i el tiempo de proceso requerido por el i-ésimo trabajo en la máquina. Supongamos que la máquina puede procesar sólo un trabajo a la vez. Además cuando un trabajo es iniciado en la máquina su proceso debe ser terminado antes de que la máquina pueda iniciar otro trabajo. Sea t_i la hora en la cual el i-ésimo trabajo es iniciado en la máquina, i = 1,...,n. El problema es determinar las restricciones sobre las variables de tiempo t_i's de tal manera que representen una secuencia implementable sobre la máquina.

Si el trabajo i es iniciado en la máquima antes que el trabajo j, debemos tener $t_j >= t_i + p_i$. Por otra parte, si el trabajo j es iniciado en la máquina antes que el trabajo i, debemos tener $t_i >= t_j + p_j$. Entonces, para cada uno de los n(n-1)/2 pares posibles i,j, i <> j, exactamente una desigualdad del par de restricciones

$$t_i \geq t_j + p_j$$

$$\text{ó} \quad t_j \geq t_i + p_i$$

debe cumplirse. Sea M un número positivo muy grande. Definamos:

$$y_{ij} = 1 \quad \text{si el trabajo } i \text{ es iniciado en la máquina antes que el trabajo } j$$

$$= 0 \quad \text{en otro caso}$$

Consideremos el sistema de restricciones:

$$My_{ij} + t_i - t_j \geq p_j$$

$$M(1 - y_{ij}) + t_j - t_i \geq p_i$$

$$y_{ij} = 0 \text{ ó } 1 \text{ para toda } i, j$$

Si (t,y) es factible para este sistema, el vector tiempo puede ser implementado en la máquina, inversamente, para cada vector t de tiempo factible, existe una y tal que el (t,y) es factible para el anterior sistema.

10) **Problema de la Mochila.**

Este modelo se refiere a la siguiente clase de problemas: artículos de n diferentes tipos están disponibles. Cada artículo de tipo i tiene un peso de w_i kg. y un valor de v_i pesos. Una mochila que puede soportar a lo más un peso de w kg. tiene que ser cargada con estos artículos de manera que se maximice el valor total de los artículos incluidos, sujeto a la capacidad de peso de la mochila. Los artículos no pueden ser divididos; únicamente un número entero no-negativo de artículos de cada tipo puede ser cargado.

Sea x_j el número de artículos de tipo j incluidos en la mochila. El problema es:

$$\text{Maximizar} \sum_{j=1}^{n} v_j x_j$$

s.c

$$\sum_{j=1}^{n} w_j x_j \leq w$$

$$x_j \geq 0 \text{ y entera para toda } j$$

Los problemas de la mochila son problemas enteros puros de una sola restricción.

1.4 EQUIVALENCIA DEL PROBLEMA ENTERO CON EL PROBLEMA 0-1.

Los ejemplos mostrados anteriormente muestran la gran variedad de problemas que pueden ser formulados como problemas de programación entera, mostraremos ahora que todo problema puro de programación entera puede ser formulado como un problema 0-1 puro.

Podemos suponer que la variable x_k posee una cota superior u_k, ya que en la realidad no es posible que una variable tome un valor por arriba de toda cota debido a que la disponibilidad de los recursos es finita lo cual trae consigo que los niveles de las actividades lo sean también. Entonces

$$x_k = \sum_{p=0}^{K} 2^p y_{kp}, \qquad y_{kp} = 0 \text{ ó } 1.$$

donde K es tal que $2^{K+1} >= u_k + 1$. Al sustituir x_k en términos de y_{kp}

tenemos el resultado deseado.

La desventaja obvia de la anterior tranformación es que el número de variables binarias puede resultar demasiado grande para permitir la resolución del problema en una cantidad de tiempo aceptable.

2 ENUMERACION IMPLICITA.

2.1 CONCEPTO DE ENUMERACION IMPLICITA.

En general, se puede suponer que el espacio de soluciones de un P.P.L.E. posee un número finito de puntos factibles. Un método directo para resolver problemas enteros es enumerar exhaustivamente (o explícitamente) todos los puntos del espacio de soluciones. En este caso, la solución óptima está determinada por el (los) punto(s) que produce(n) el mejor (máximo ó mínimo) valor de la función objetivo.

La desventaja obvia de la técnica anterior es que el número de puntos solución puede ser demasiado grande para fines prácticos, con el resultado de que la solución no puede ser determinada en una cantidad razonable de tiempo. La idea de la enumeración implícita (o parcial) es la de considerar solamente una porción (posiblemente pequeña) de todos los posibles puntos solución y descartar los restantes como no-prometedores. Para ilustrar este punto, consideremos el problema de determinar todas las soluciones factibles para la siguiente desigualdad:

$$3x_1 - 8x_2 + 5x_3 \leq -6, \qquad x_j = (0,1), \quad j = 1,2,3$$

Por simple inspección vemos que para cualquier solución factible x_2 debe ser fijada a nivel 1. Esto significa que cualquier combinación binaria (x_1,x_2,x_3) teniendo $x_2 = 0$ no puede producir una solución factible. Por lo tanto las cuatro combinaciones (0,0,0), (1,0,0), (0,0,1) y (1,0,1) son descartadas automáticamente como no-prometedoras y se dice que están implícitamente enumeradas.

Dado que $x_2 = 1$ es un requerimiento necesario para tener factibilidad, la holgura de la desigualdad está dada por -6-(-8) = 2 unidades. Por lo tanto, x_1 o x_2 (ó ambas) pueden tomar valor 1 sólo si sus contribuciones al lado izquierdo de la desigualdad no excede 2. Ya que el coeficiente de x_1 es igual a 3, x_1 debe ser fijada a nivel 0, lo cual significa que (1,1,0) y (1,1,1) son implícitamente enumeradas como no-prometedoras. Ahora, dado que $x_1 = 0$ y $x_2 = 1$, la combinación (0,1,1) es descartada ya que el coeficiente de x_3 es igual a 5. La combinación restante (0,1,0) es la única solución factible a la desigualdad.

Para apreciar el impacto de utilizar la enumeración implícita, la figura 2.1 da una representación gráfica de las ideas utilizadas en el ejemplo anterior. La primera conclusión es que x_2 debe ser fijada a nivel 1. Entonces todas las ramas que surgen de la rama $x_2 = 0$ (mostradas con línea punteada) son descartadas. En este caso, se dice que $x_2 = 0$ está sondeado. Para $x_2 = 1$, la desigualdad muestra que $x_1 = 1$ no puede producir una solución factible y por lo tanto ($x_2 = 1, x_1 = 1$) está sondeado. Ahora, dado $x_2 = 1$ y $x_1 = 0$, la rama $x_3 = 1$ conduce a una solución infactible, pero la rama $x_3 = 0$ conduce a una solución factible. Ya que todas las combinaciones (2^3) han sido consideradas, el proceso termina.

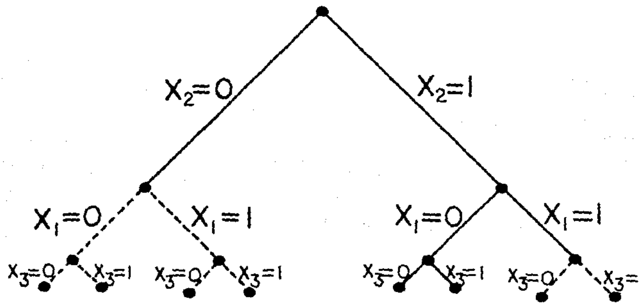


fig. 2.1

De la discusión anterior podemos obtener dos puntos importantes para la exitosa implementación de la enumeración implícita:

(1) Existe la necesidad de un esquema de enumeración que asegure que todos los puntos solución son enumerados (implícita o explícitamente) de una forma no-redundante.

(2) Un número de pruebas "sondeadoras" deben ser diseñadas para excluir tantas soluciones no-prometedoras como sea posible.

En el ejemplo anterior es fácil no perder de vista las soluciones enumeradas porque hay tan sólo ocho de ellas. En el caso general, un método de enumeración eficiente y flexible es requerido para no perder de vista todas las soluciones que han sido consideradas (ya sea implícita o explícitamente) y para generar las restantes de una forma no-redundante. En donde se entiende por eficiente que el método de enumeración no sobrecargue la memoria de computadora y por flexible que guarde la información y que tenga acceso a ella fácilmente.

Las pruebas sondeadoras provistas para el ejemplo anterior están especialmente diseñadas para tomar ventaja de su estructura. Para el caso general, estas pruebas deben ser diseñadas para cubrir una clase más amplia de problemas. Debe observarse que las pruebas sondeadoras pueden conducir a un algoritmo potente o débil dependiendo principalmente de cuán inteligentemente estas pruebas sean diseñadas para detectar las soluciones no-prometedoras.

En el presente trabajo se proveerán once de estas pruebas las cuales se podrán utilizar en el orden que se deseé, lo cual permite el estudio de diferentes secuencias en el orden de aplicación pudiéndose con ello desarrollar algoritmos y estudiar la eficiencia de éstos.

2.2 ESQUEMA DE ENUMERACION.

La enumeración implícita no considera combinaciones binarias "completas". En cambio, ésta empieza con una (o más) variables fijadas a un valor binario (combinación binaria parcial) y entonces, gradualmente se construye la "solución" al aumentar (a la combinación binaria parcial) variables a un valor fijo. Como ilustración, el ejemplo de enumeración de la sección 2.1 empieza por fijar x_2 a nivel uno, entonces x_1 es aumentada (a la combinación binaria parcial) a nivel cero, y así sucesivamente. En el curso de cada una de estas argumentaciones es evidente que puntos solución (asignaciones completas) pueden ser descartados sin ser considerados explícitamente. Esta idea es la base para el esquema de enumeración que utilizaremos, el cual fue diseñado por Glover.

Se dan las siguientes definiciones para facilitar la presentación del esquema. Por conveniencia, la notación +j (-j) se usa para indicar que $x_j = 1$ ($x_j = 0$). Por ejemplo, la información $x_5 = 0$, $x_2 = 1$ y $x_3 = 1$ puede ser resumida como {-5, +2, +3}.

(i) **Solución Parcial (G)** Es un conjunto ordenado, el cual asigna especificaciones binarias a un subconjunto G de N. En el ejemplo de la sección 3.1,el conjunto G = {+2,-1} es una solución parcial en la cual $x_2 = 1$ y $x_1 = 0$.

(ii) **Variables Libres (F = N - G)** Una variable que no está asignada a un valor binario por la solución parcial G es libre de tomar valor cero o uno.

(iii) **Variables con Valor 0 (X)** Son las variables a las cuales se les ha asignado (de manera temporal o definitiva) el valor cero.

(iv) **Variables con Valor 1 (L = N - (F + X))** Son las variables a las cuales se les ha asignado (de manera temporal o definitiva) el valor uno.

(v) **Completación de G** Se asigna un valor binario a todas las variables en N, estos valores están determinados por la asignación en G junto con una especificación binaria para cada variable libre. En el ejemplo anterior, G = {+2,-1} tiene dos completaciones: {+2,-1,-3} y {+2,-1,+3}.

(vi) **Variable Cancelada** Es una variable a la cual se le ha asignado un valor binario (cero o uno) definitivamente en la solución parcial y se **conmemorará** mostrando en letras negritas subrayadas el elemento correspondiente en la asignación parcial.
Por ejemplo, en la asignación parcial (+2,**-1**) la variable x_1 ha tomado definitivamente el valor cero, es decir que no le asignaremos el valor uno (esto es práctico para indicar que todas las completaciones de la asignación parcial (+2,+1) ya han sido sondeadas de manera que no se volverán a generar estas completaciones).

(iv) **Solución parcial sondeada** Se dice que una solución parcial G está sondeada si todas sus completaciones pueden ser descartadas como no-prometedoras. En el ejemplo de la sección 3.1, G = (-2) y G = (+2,+1) son soluciones parciales sondeadas. (Ver fig. 2.1).

El diagrama de flujo para el esquema de enumeración de Glover se muestra en la figura 2.2. La mecánica del esquema se muestra primero utilizando el ejemplo dado en la sección 2.1. La figura 2.3 ilustra los pasos gráficamente de manera que se puede apreciar el efecto del esquema comparado con el árbol que muestra la enumeración completa en la figura 2.2. Para propósito del ejemplo, una solución parcial se dice sondeada sólo si no tiene completaciones factibles.

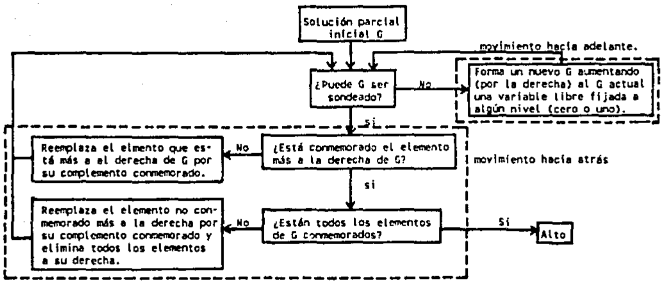


Figura 2.2 Diagrama de flujo del Esquema de Enumeración de Glover

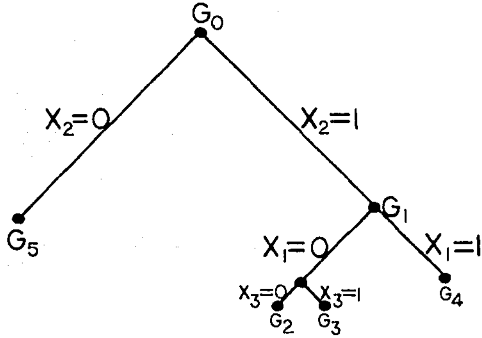


fig. 2.3

Supongamos que la solución parcial inicial es vacía, ésto es, $G_0 = \{\}$. Entonces, las variables x_1, x_2 y x_3 son libres. Ya que $x_2 = 1$ es un paso prometedor para alcanzar factibilidad, el **movimiento**

hacia adelante (ver el diagrama de flujo) es ejecutado para producir $G_1 = (+2)$. Sin embargo, ya que el coeficiente de x_2 es menor que el lado derecho por sólo 8 - 6 = 2, la factibilidad puede ser alcanzada al fijar los valores de las dos variables x_1 y x_2 a nivel cero. Por lo tanto el movimiento hacia adelante produce $G_2 = (+2,-1,-3)$ y G_2 es etiquetado como una solución factible. Ya que G_2 es una asignación binaria completa, ésta es obviamente sondeada. De acuerdo con la figura 3.2, el **movimiento hacia atrás** produce $G_3 = (+2,-1,\underline{\pm 3})$. El número en negritas subrayado **conmemora** que G_2 (en el cual $x_3 = 0$) ha sido sondeado. G_3 (el cual es infactible) es una asignación completa y por lo tanto está sondeada. Entonces, el movimiento hacia atrás produce $G_4 = (+2,\underline{\pm 1})$. Ya que el coeficiente de la variable x_3 es positivo, G_4 no tiene una completación factible y por lo tanto está sondeado. Esto produce $G_5 = (\underline{-2})$ el cual está sondeado ya que $x_2 = 1$ es una condición necesaria para tener factibilidad. Como todos los elementos de G_5 (el cual ya está sondeado) están conmemorados, la enumeración está completa.

Si supiésemos desde el principio que x_2 debe ser igual a uno en cualquier solución factible de la desigualdad, la secuencia de soluciones factibles sería la siguiente:

$$G_1 = (\underline{\pm 2})$$

$$G_2 = (\underline{\pm 2},-1,-3)$$

$$G_3 = (\underline{\pm 2},-1,\underline{\pm 3})$$

$$G_4 = (\underline{\pm 2},\underline{\pm 1})$$

Como G_4 está sondeado, y ya que todos sus elementos están conmemorados, la enumeración está completa y el proceso termina. La idea es que el conmemorado en $G_1 = (\underline{\pm 2})$ es equivalente a decir que la solución parcial (-2) ha sido sondeada, es decir, que las completaciones en las cuales $x_2 = 0$ no pueden conducir a una solución factible.

El esquema de enumeración de Glover explora las soluciones parciales en base a la regla LIFO (last-in,first-out). Esto surge de la manera en que son generadas las soluciones parciales. Por ejemplo, en la figura 2.3, cuando G_1 es considerado para ser sondeado, el esquema (implícitamente) guarda G_5 para posterior exploración. Además cuando G_2 es considerado para ser sondeado, G_4 y G_3 son guardados para posterior exploración. En este punto de la enumeración, la lista incluye G_5, G_4 y G_3. Aunque, de acuerdo al esquema, G_3 es el último en ser guardado es el primero en ser sacado de la lista para ser explorado.

Las 8 ($=2^3$) combinaciones del ejemplo numérico anterior han sido enumeradas (implícita o explícitamente). De acuerdo a la figura 2.3, las soluciones parciales sondeadas son G_2, G_3, G_4 y G_5. Observemos que en un problema de n variables el número de completaciones de una solución parcial de s elementos es 2^{n-s}, ya que tenemos (n - s) variables libres y cada una de ellas puede tomar dos valores. El número de completaciones enumeradas en el ejemplo anterior es $2^{3-3} + 2^{3-3} + 2^{3-2} + 2^{3-1} = 8$.

De todo lo anterior se puede observar que:

(i) Todas las futuras completaciones no duplicarán a ninguna de las precedentes.

(ii) El esquema de enumeración finaliza sólo hasta que todas las posibles soluciones (2^n) han sido enumeradas (implícita o explícitamente).

2.3 PRUEBAS SONDEADORAS

Antes de discutir las pruebas sondeadoras (criterios de enumeración implícita y de ramificación) es necesaria alguna notación. Consideremos el problema general de programación lineal entera puro

$$\begin{aligned} \text{Minimizar} \quad & \boldsymbol{c}\boldsymbol{x} = z \\ \text{sujeto a} \quad & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & 0 \leq \boldsymbol{x} \leq \boldsymbol{e} \\ \text{con} \quad & x_j = 0 \text{ o } 1 \text{ para } j = 1,\dots,n; \end{aligned}$$

sea el **subproblema en el nodo $\mathbf{x}^l$**

$$\begin{aligned} \text{Minimizar} \quad & \boldsymbol{c}\boldsymbol{x}^l = z \\ \text{sujeto a} \quad & \boldsymbol{A}\boldsymbol{x}^l \leq \boldsymbol{b} \\ & 0 \leq \boldsymbol{x}^l \leq \boldsymbol{e} \\ \text{con} \quad & x_j = 0 \text{ o } 1 \text{ para } j = 1,\dots,n; \end{aligned}$$

en donde $\mathbf{A}$ es una matriz de m por n, $\mathbf{c}$ es el renglón de costos, $\mathbf{b}$ es un vector de constantes y $\mathbf{e}$ es una columna de unos. El punto $\mathbf{x}^l$ contiene l variables asignadas (con valor 0 o 1) y $f = n - l$ variables libres. Al hacer las operaciones indicadas y simplificar tenemos el siguiente problema equivalente:

$$\begin{aligned} \text{Minimizar} \quad & \boldsymbol{c}^f\boldsymbol{x}^f = z - z^l \\ \text{sujeto a} \quad & \boldsymbol{A}^f\boldsymbol{x}^f \leq \boldsymbol{b}^l \\ & 0 \leq \boldsymbol{x}^f \leq \boldsymbol{e}^f \\ \text{con} \quad & x_j = 0 \text{ o } 1 \text{ para } j \in F; \end{aligned}$$

en donde $\mathbf{x}^f = (x_j)$ es el vector correspondiente a las variables libres, F es el conjunto correspondiente a los índices en $\mathbf{x}^f$, c^f y $\mathbf{A}^f = (\mathbf{a}_j)$ son los costos y las columnas de A asociadas a los índices en $\mathbf{x}^f$, a_{ij} es la i-ésima componente de la columna $\mathbf{a}_j$, $\mathbf{b}^l = (b_{il})$ es la columna b actualizada y z^l es la suma de los costos de las variables asignadas a 1. Denotaremos a este subproblema por P^l. El valor de cualquier solución a P^l incluirá el costo z^l de las variables fijas (asignadas a algún valor). El problema de programación lineal (sin el requerimiento de que las variables sean enteras) será denotado por PL^{l} y su solución óptima es denotada por $z^{\wedge}$, la cual incluye a la constante z^{l}. La solución entera actual mínima (la cual inicialmente será la cota superior dada por la suma de los costos más uno ya que estamos trabajando con un P.P.L.E con $c >= 0$) es denotada por z^{*}.

La teoría se desarrollará (sin pérdida de generalidad) suponiendo que se tiene un P.P.L.E de minimización con vector c de costos no-negativo. En el caso en que el P.P.L.E. tenga al menos un coeficiente de costo negativo, se puede transformar el problema en uno equivalente con vector de costos no-negativo mediante el siguiente cambio de variables (recuérdese que son variables 0-1).

$$x_j' = 1 - x_j \quad \text{si } c_j < 0$$

$$x_j' = x_j \qquad \text{si } c_j \geq 0$$

Si el problema es de maximización, se transforma éste en uno equivalente de minimización a través de la ecuación max $z = -\min -z$.

Una observación importante es la siguiente: si el P.P.L.E. original es uno de maximización y c tiene al menos un elemento negativo, se debe dar preferencia al cambio por una función objetivo a minimizar y luego realizar el cambio de variables señalado (si es necesario). Por ejemplo:

Sea Max $Z = 3x_1 + 2x_2 - 3x_3 - 8x_4$

S.C.

$$x_1 + x_2 + 3x_3 - x_4 \leq 3 \quad \text{Problema}$$

$$-x_1 + 2x_2 + 4x_3 + x_4 \leq 7 \quad \text{Original}$$

$$x_i = (0,1)$$

el problema con función a minimizar sería:

$$\text{Min } Z = -3x_1 - 2x_2 + 3x_3 + 8x_4$$

S.C.

$$x_1 + x_2 + 3x_3 - x_4 \le 3$$
$$-x_1 + 2x_2 + 4x_3 + x_4 \le 7$$
$$x_i = (0,1)$$

el problema con función a minimizar y vector de costos no negativo sería:

$$\text{Min } Z = 3x_1' + 2x_2' + 3x_3 + 8x_4 - 5$$

S.C.

$$-x_1' - x_2' + 3x_3 - x_4 \le 1 \quad \text{Problema}$$
$$x_1' - 2x_2' + 4x_3 + x_4 \le 6 \quad \text{Modificado}$$
$$x_i = (0,1) \quad i \in \overline{3,4}$$
$$x_i' = (0,1) \quad i \in \overline{1,2}$$

Antes de aplicar cada prueba se verifica si el problema en el nodo $\mathbf{x}^l$ (más brevemente el nodo $\mathbf{x}^l$, el nodo actual) puede producir una mejor solución y en caso afirmativo se probará si la completación trivial (asignar todas las restantes variables libres a cero) produce una mejor solución entera que la actual; ésto se hace con el siguiente razonamiento:

Si $z^l > z^*$ entonces el nodo $\mathbf{x}^l$ no puede producir una mejor solución que la actual ya que $c >= 0$ y se efectuará un movimiento hacia atrás (por medio de la regla LIFO).

Si $z^l = z^*$ y tenemos que $b^l >= 0$ (i.e., tenemos factibilidad con la completación trivial) entonces se ha encontrado una solución óptima alternativa.

Por otra parte, si

$$z^l < z^*.$$

y tenemos que $b^l >= 0$ (i.e., tenemos factibilidad con la completación trivial) entonces se ha llegado a una mejor solución entera.

Supongamos ahora que después de alcanzar el nodo $\mathbf{x}^l$ en un movimiento hacia adelante, una mejor solución entera es encontrada. Entonces, z^* es reducida a z^l y los nodos adicionales definidos a partir de $\mathbf{x}^l$ no podrán producir nunca una solución entera por debajo de este valor (ya que los coeficientes de costo son no-negativos); de esta manera, obtener factibilidad en un nodo (i.e., $b^l >= 0$) con $z^l < z^*$ siempre traerá consigo un inmediato movimiento hacia atrás.

En caso de que se obtenga una solución óptima alternativa un movimiento hacia atrás está justificado.

2.3.1 PRUEBA CEILING.

En el caso en que b^l tenga al menos un elemento negativo (i.e., la completación trivial no es factible) y exista una variable x_j libre cuyo coeficiente es tal que:

$$c_j + z^l > z*$$

entonces x_j debe ser cancelada a nivel cero.

2.3.2 PRUEBA INFACTIBILIDAD.

Las restricciones del subproblema actual, con variables de holgura $\mathbf{s}^l = (s_i)$, pueden ser escritas como:

$$\mathbf{s}^l = \mathbf{b}^l - \mathbf{A}^f \mathbf{x}^f \geq 0$$

Ya que x_j no debe exceder de 1, el valor más grande posible para cada variable de holgura s_i es:

$$P_i = b_{il} - \sum_{j \in F^-} a_{ij},$$

$$\text{con } F^- = \{j \in F \mid a_{ij} < 0\}$$

Por lo tanto, para que una solución cero-uno sea posible, debemos de tener que:

$$P_i \geq 0 \quad \text{para} \quad i = 1,\dots,m. \tag{1}$$

Observemos que $P_i >= b_{ii}$ y que por lo tanto, para $b_{ii} >= 0$ la última desigualdad es automáticamente satisfecha. Concluímos entonces que, calcularemos P_i sólo para cada restricción con $b_{ii} <$ 0. Si uno o más de esos valores es negativo, un paso hacia atrás está justificado.

2.3.3 PRUEBA CERO.

Supongamos que una variable libre tiene un coeficiente positivo (grande) en algún renglón. Entonces, el asignar a esta variable el valor 1 reduce b_{ii} pudiendo resultar que $P_i < 0$. Entonces, si para cualquier j en F

$$P_i - \underset{a_{ij} > 0}{\text{máximo}}\ a_{ij} < 0 \quad \text{para alguna } i \quad (1 \le i \le m), \qquad (2)$$

cancelamos x_j a nivel 0.

Notemos que si la desigualdad anterior indica una cancelación, una columna $\mathbf{a}_j$ será omitida de A^f y en consecuencia los valores de P_i pueden alterarse. Así, después de cada cancelación, P_i debe ser recalculado y volverse a probar (1) y (2). El siguiente ejemplo ilustra la continua recalculación e interacción entre las pruebas Infactibilidad y Cero.

Ejemplo: Consideremos que las restricciones i y j aparecen en un subproblema

$$i : -4x_3 + 50x_6 + x_7 \le -3$$

$$j : 20x_3 - 20x_6 - x_7 \le 5$$

En nuestra notación, F = {3,6,7}, P_i = -3 - (-4) = 1, y P_j = 5 - (-20 -1) = 26; así, es posible (a primera vista) satisfacer las restricciones i y j. Pero P_i = 1 y

$$1 - \underset{a_{ij} > 0}{\text{máximo}}\ a_{ij} = 1 - 50 = 0 < 0,$$

y por lo tanto cancelamos x_6, (claramente, el asignar x_6 a 1 hace imposible el satisfacer la desigualdad i teniendo x_3 con valor a lo más uno). Realizando la cancelación, tenemos que P_i = 1 y que P_j se reduce a 6 = 5 - (-1). El probar (2) con el renglón actualizado j : $20x_3 - x_7 <= 5$ nos permite cancelar x_3, pero entonces P_i = -3 < 0 y un paso hacia atrás está justificado. Después de cancelar x_3 y x_6 con valor 0, la restricción i : $x_7 <= -3$ es obviamente inconsistente con $x_7 = 0$ ó 1.

2.3.4 PRUEBA UNO

El número P_i es la suma de b_{il} y el negativo de la suma de los coeficientes negativos (en el renglón i) de las variables libres. Supongamos que uno de esos coeficientes fue omitido cuando calculamos P_i (ie, una variable libre fue temporalmente asignada a 0). Si ésto resulta en $P_i < 0$ entonces, para que sea posible producir una solución cero-uno la variable omitida debe tomar valor 1. Por lo tanto, si para alguna j en F

$$P_i + \underset{a_{ij}<0}{\text{mínimo}}\ a_{ij} < 0 \quad \text{para alguna } i \quad (1 \le i \le m), \tag{3}$$

entonces x_j debe tomar valor 1 en cualquier solución cero-uno producida a partir del nodo $\mathbf{x}^l$. Si $b_{il} >= 0$, observemos que $P_i + a_{ij} >= b_{il} >= 0$ para cualquier $a_{ij} < 0$, es decir, que (3) nunca indicará que una variable debe tomar valor 1, en otras palabras, la prueba se realizará para aquellos renglones con $b_{il} < 0$.

Ejemplo: Supongamos que el renglón i es:

$$i : 5x_2 - 2x_3 - 4x_4 - x_6 \le -5$$

entonces $P_i = -5 - (-2 - 4 - 1) = 2 > 0$ y $2 + (-4) < 0$: por lo tanto x_4 debe tomar valor 1 en cualquier solución cero-uno encontrada a partir de $\mathbf{x}^l$; es decir, cancelamos x_4 con valor 1 en el nodo 1.

2.3.5 PRUEBA PROGLIN

El subproblema es denotado por P^l y su problema de programación lineal asociado por PL^l. Ya que el conjunto de puntos cero-uno que satisfacen las restricciones de P^l está contenido en la región de soluciones factibles definidas por las restricciones de PL^l, las siguientes observaciones son evidentes:

i) Si la solución óptima a PL^l es entera (en las variables libres), entonces esta solución es óptima para P^l.

ii) Si PL^l no tiene soluciones factibles, entonces no existe ninguna solución cero-uno para P^l.

iii) El valor óptimo z^* de la función objetivo para el problema continuo es una cota inferior sobre el valor z para cualquier solución cero-uno al subproblema; ésto es, $\hat{z} <= z$.

Supongamos que PL^l es resuelto por el algoritmo dual-simplex para variables acotadas (la base inicial está constituida por las variables de holgura ya que $c >= 0$ y por lo tanto $z_j - c_j < 0$, ya que estamos minimizando se cumple el criterio de optimalidad primal por lo que la base es dual-factible). Entonces el valor de la función objetivo es no-decreciente, y podríamos utilizar (iii) para derivar una prueba de cota la cual nos puede indicar un paso hacia atrás antes de terminar los cálculos del algoritmo dual-simplex para variables acotadas. Esto es:

iv) Si en cualquier iteración del algoritmo dual-simplex para variables acotadas el valor z de la función objetivo del problema lineal excede el valor de la mejor solución entera actual z^*, entonces un paso hacia atrás está justificado.

2.3.6 PRUEBA PENALIDADES

Después de resolver el problema continuo PL^l (por medio del algoritmo dual-simplex para variables acotadas), podemos considerar el efecto en el valor mínimo $\hat{z}$ de la función objetivo provocado por forzar a cada variable entera a tomar valor 0 o 1 (una por una individualmente). Evidentemente tal requerimiento impone una restricción adicional al problema, y después de reoptimizar, un nuevo valor de la función objetivo $z_{min} >= \hat{z}$ es obtenido. La diferencia $D = z_{min} - \hat{z}$ es la penalidad por forzar a la variable a tomar valor 0 o 1. Desde el punto de vista computacional, puede resultar muy costoso determinar la penalidad D de manera exacta para cada variable por lo que se estimará una cota inferior d para el valor D lo cual reducirá substancialmente el requerimiento de memoria de la computadora. Aunque $\hat{z} + d$ no es tan fuerte como $\hat{z} + D$, es evidente que si $\hat{z} + d > z^*$ entonces, la variable bajo consideración debe ser cancelada a 0 o 1 dependiendo si ésta fue forzada a tomar valor 1 o 0 respectivamente. En el caso en que una variable con valor fraccional (ie, básica) tenga penalidades d_0 y d_1 por forzarla a tomar valor 0 y 1 respectivamente con la característica que $\hat{z} + d_0 > z^*$ y $\hat{z} + d_1 > z^*$ entonces, un paso hacia atrás está justificado. Podemos resumir lo dicho anteriormente en la tabla 2.1

Tabla 2.1

Valor Actual de la Variable x_j	Restricción Adicional $x_j = 0$	$x_j = 1$	Resultado cuando $\hat{z} + d > z^*$
0	--	▲	x_j es cancelada a 0
1	▲	--	x_j es cancelada a 1
fraccional	--	▲	x_j es cancelada a 0
fraccional	▲	--	x_j es cancelada a 1
fraccional	▲	▲	paso hacia atrás

▲ Significa que se impuso la restricción y $\hat{z} + d > z^*$

- **Cálculo de las Penalidades:**

En la tabla óptima, alguna de las variables binarias x_j son básicas tomando valor fraccional o entero. Las restantes variables x_j son no-básicas tomando valor 0 (en su cota inferior) o valor 1 (en su cota superior). Esto es debido a que la restricción $0 <= x_j <= 1$ es implícita. Por lo tanto, al estimar la penalidad d, debemos tomar en cuenta dos tipos de restricciones enteras:

i) Una variable no-básica x_j con valor cero (uno) es forzada a tomar valor uno (cero).

ii) Una variable fraccional $x_j = x_j^*$, $0 <= x_j^* <= 1$, es forzada a tomar un valor binario. Un caso especial del tipo ii) ocurre cuando una variable básica x_j toma valor entero pero es forzada a tomar el otro valor entero.

Supongamos que la ecuación para la función objetivo en la tabla óptima del problema continuo está escrita de la siguiente manera:

$$z + \sum_{k \in NB} (z_k - c_k)\, x_j = z^*$$

en donde NB es el conjunto formado por las variables no-básicas y $z_k - c_k$ son los coeficientes de costo reducido para las variables no-básicas.

De la teoría de la programación lineal, sabemos que la función objetivo es una función convexa lineal por segmentos. Una ilustración típica es mostrada en la figura 2.4

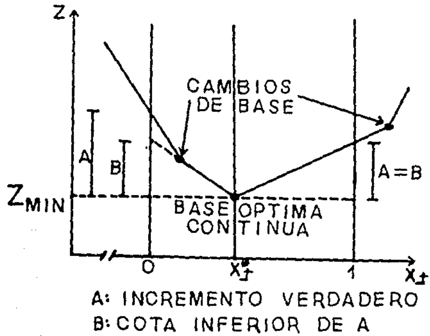


figura 2.4

Debido a la convexidad de la función alrededor de x_j^*, el incremento verdadero en el valor óptimo de z como resultado de asignar un valor binario a x_j es al menos igual al incremento calculado cuando $z_k - c_k$, k en NB, permanece sin cambiar, ésto es, cuando no se da un cambio de base. Esta es la idea para calcular las penalidades como serán ahora detalladas.

i) **Penalidades para Variables No-Básicas:** De la fórmula

$$z = z^* - \sum_{j \in NB} (z_j - c_j)\, x_j$$

tenemos que la penalidad por incrementar (decrementar) la variable binaria x_j a nivel uno (cero) es directamente igual a $-(z_j - c_j)$ ya que estamos suponiendo que no se da un cambio de base.

ii) **Penalidades para Variables Básicas:** Expresemos la restricción asociada con la variable básica x_j en la tabla óptima en la forma:

$$x_j = x_j^* - \sum_{k\in NB} y_{jk}x_k, \qquad 0 < y_j^* < 1$$

el que x_j sea decrementada a nivel cero es equivalente a imponer la restricción $x_j <= 0$ en la solución óptima continua. De la anterior restricción, $x_j <= 0$ es equivalente a:

$$-\sum_{k\in NB} y_{jk}x_k \le -x_j^*$$

Similarmente, el forzar a x_j a nivel uno es equivalente a imponer la restricción $x_j >= 1$, ie

$$\sum_{k\in NB} y_{jk}x_k \le x_j^* - 1$$

Al aumentar las restricciones anteriores (una a la vez) a la tabla óptima continua y aplicando el algoritmo dual-simplex, el valor de la función objetivo debe incrementarse por lo menos en:

$$d_0 = d_{j0} = -x_j^* \max\left\{\frac{z_k - c_k}{y_{jk}} \mid y_{jk} > 0\right\}, \quad \text{si } x_j = 0$$

en donde el máximo es sobre las k en NB.

$$d_1 = d_{j1} = -(x_j^* - 1) \min\left\{\frac{z_k - c_k}{y_{jk}} \mid y_{jk} < 0\right\}, \quad \text{si } x_j = 1$$

en donde el mínimo es sobre las k en NB.

La prueba va calculando la(s) penalidad(es) para cada variable en N, finalizando cuando encuentra la primera variable que deba ser cancelada de acuerdo a la tabla 2.1.

2.3.7 PRUEBA RAM VAR MAX PEN.

Cuando el problema de programación lineal PL^l es resuelto, se pueden calcular las penalidades descritas en la prueba 2.3.6 y elegir para ramificar (ie, asignar una de las variables libres a un valor binario) la variable (básica o no-básica) con mayor penalidad. Esta elección puede conducir a una rápida cancelación de la variable seleccionada (lo cual puede causar otras cancelaciónes) y regresar rápidamente al nodo anterior. Notemos que si esta prueba selecciona una variable básica (la cual tiene dos penalidades) entonces se asignará a esta variable el valor binario correspondiente a la mayor de sus dos penalidades, así mismo, recordemos que como se está utilizando el algoritmo dual-simplex para variables acotadas para resolver PL^l la prueba podría indicar una asignación a cero de una variable no-básica ya que éstas pueden estar en su cota inferior o superior en la tabla óptima del problema continuo.

2.3.8 PRUEBA RAM VAR MEN PEN.

Cuando el problema de programación lineal PL^l es resuelto, se pueden calcular las penalidades descritas en la prueba 2.3.6 y elegir para ramificar (ie, asignar una de las variables libres a un valor binario) la variable (básica o no-básica) con menor penalidad. Es de esperar que esta elección conduzca a la enumeración a una "buena" u óptima solución. Notemos que si esta prueba selecciona una variable básica (la cual tiene dos penalidades) entonces se asignará a esta variable el valor binario correspondiente a la menor de sus dos penalidades, así mismo, recordemos que como se está utilizando el algoritmo dual-simplex para variables acotadas para resolver PL^l la prueba podría indicar una asignación a cero de una variable no-básica ya que éstas pueden estar en su cota inferior o superior en la tabla óptima del problema continuo.

Observación para las Pruebas MaxPenRam y MinPenRam.

Ya que la prueba 2.3.8 busca una solución óptima y la prueba 2.3.7 busca producir cancelaciones y pasos hacia atrás, una estrategia para resolver el P.P.L.E sería la siguiente:

Utilizar la prueba 2.3.8 hasta que se sienta que una solución óptima o cercana a la óptima ha sido encontrada y entonces cambiar hacia la prueba 2.3.7.

2.3.9 PRUEBA PREFERSET.

Supongamos que la i-ésima restricción actual está escrita en la forma:

$$\sum_{j \in F} a_{ij} x_j \leq b_{i1} \qquad (3)$$

y que tiene a $b_{i1} < 0$.

Esta desigualdad puede ser escrita como:

$$\sum_{j \in F^+} a_{ij} x_j + \sum_{j \in F^-} a_{ij} x_j \leq b_{i1} \qquad (4)$$

en donde $F^+ = \{ j \text{ en } F \mid a_{ij} > 0 \}$ y $F^- = \{ j \text{ en } F \mid a_{ij} < 0 \}$.

Multiplicando esta última expresión por -1 y rearreglando términos, tenemos:

$$\sum_{j \in F^-} (-a_{ij})\, x_j \geq -b_{i1} + \sum_{j \in F^+} a_{ij} x_j \geq -b_{i1} \qquad (5)$$

Una vez obtenida esta expresión, aplicamos el proceso de **reducción completa** a esta desigualdad, el cual será mucho mejor entendido a través del siguiente ejemplo:

Supongamos que el renglón i es:

$$5x_2 - 2x_3 - 4x_4 - x_6 \leq -5 \qquad (3)'$$

ésto implica que:

$$2x_3 + 4x_4 + x_6 \geq 5 + 5x_2 \geq 5 \qquad (5)'$$

ya que x_6 no puede exceder de 1, tenemos que:

$$2x_3 + 4x_4 \geq 5 - x_6 \geq 4$$

además, $x_3 <= 1$, lo cual nos conduce a:

$$4x_4 \geq 4 - 2x_3 \geq 2$$

lo cual significa que:

$$x_4 \geq \frac{1}{2} > 0$$

lo que significa que x_4 debe tomar valor 1.

El proceso de reducción completa substrae de (5) el término a_{ij} ($a_{ij} < 0$) por orden decreciente de los a_{ij} hasta el momento en que la siguiente substracción volvería a b_{il} no positiva. La desigualdad resultante tiene la forma:

$$\sum_{j \in P} x_j \geq r \qquad \text{con } P \subset F^- \subset F, \quad r > 0 \qquad (6)$$

a P se le conoce como el conjunto preferido y a una x_j en P se le conoce como **variable preferida**.

Es claro que sólo es necesario considerar a las variables preferidas como candidatas a ramificar. La idea es encontrar el conjunto preferido con la menor cardinalidad y elegir una de las variables en este conjunto para ser asignada (no cancelada) a valor 1. La manera de encontrar el conjunto preferido con la menor cardinalidad es efectuar el proceso de reducción completa a cada restricción con $b_{il} < 0$ y seleccionar el conjunto preferido con el menor número de elementos. Notemos que si el conjunto encontrado es de cardinalidad uno, entonces se deberá cancelar a esta variable con valor 1.

2.3.10 PRUEBA BALAS.

Si a una variable libre x_j se le asigna el valor 1, entonces el término constante b_{il} en cada restricción i se transforma en $b_{il} - a_{ij}$. Cuando $b_{il} - a_{ij} >= 0$, la restricción i es satisfecha, y por lo tanto una medida de la "infactibilidad de la restricción" (con $x_j = 1$) está dada por:

$$v_j = \sum_{i \in M_j} (b_{il} - a_{ij}), \qquad M_j = \{i | (b_{il} - a_{ij}) < 0, i = 1, \ldots, m\}$$

definimos $v_j = 0$ siempre que $M_j = \{\}$.

Lo que esta prueba hace es calcular el conjunto preferido con la menor cardinalidad (en la forma descrita en 2.3.9) y para cada variable preferida calcula la penalidad v_j y elige para ramificar a la variable con la mayor penalidad. Observemos que $v_j <= 0$, y que cuando $v_j = 0$ se ha encontrado una solución fijando a x_j con valor 1 y las restantes variables con valor 0.

Ejemplo: Consideremos las restricciones:

$$-3x_{10} - x_{12} - 5x_{14} \leq -1,$$

$$8x_{10} - 3x_{12} + x_{14} \leq 5,$$

$$-3x_{10} - 4x_{12} - x_{14} \leq -4.$$

F = {10,12,14}. El aplicar el proceso de reducción completa al primer renglòn produce el conjunto preferido {10,12,14}, y aplicándolo al tercer renglón obtenemos el conjunto preferido {10,12}. Calculando v_j para las variables en el conjunto preferido con la menor cardinalidad obtenemos:

$$v_{10} = -3 - 1 = -4$$

$$v_{12} = 0;$$

de manera que x_{12} es cancelada con valor 1 lo cual produce que las restricciones se satisfagan con $x_{10} = x_{14} = 0$.

2.3.11. PRUEBA SURROGATE.

Notemos que en las pruebas 2.3.3. y 2.3.4 las restricciones son utilizadas una por una por una. Glover observó que se puede obtener mejor información acerca de la factibilidad y optimalidad del problema relativo a sus variables libres, si las restricciones son "combinadas" en una desigualdad, la cual se supone delimita mejor el espacio de soluciones factibles. La desigualdad, a la cual nos referiremos como la **restricción substituta**, está formada por una combinación lineal no-negativa de las restricciones originales del problema. En otras palabras, dado un problema 0-1 en forma matricial como:

$$\min\{CX \mid AX \leq b,\ x_j = (0,1),\ j \in N\}$$

una restricción substituta es definida como:

$$u(AX - b) \leq 0$$

donde u es un vector no-negativo. En general, la restricción substituta está definida con respecto a la solución parcial actual G. Esto significa que antes de que la restricción substituta sea construída, las variables designadas por la solución parcial son substituídas en las restricciones originales con sus valores binarios especificados y por lo tanto, la restricción substituta queda únicamente definida en términos de las variables libres.

Para ilustrar la utilidad de la restricciones substitutas, consideremos las dos desigualdades, $-x_1 + 2x_2 <= -1$ y $2x_1 - x_2 <= -1$. Por separado, cada restricción tiene una solución factible. Sin embargo,cuando son consideradas simultáneamente, el sistema es infactible. Este resultado puede ser obtenido al sumar las dos restricciones, ésto es, $u = (1,1)$, lo que produce la restricción substituta $x_1 + x_2 = -2$; esta restricción claramente muestra la infactibilidad del sistema.

Glover demuestra dos propiedades importantes de la anterior restricción substituta:

(i) Si $\underline{X}$ es una solución factible al problema original, entonces $\underline{X}$ es factible para la restricción substituta.

(ii) Si la restricción substituta no tiene solución factible, entonces el problema original tampoco tiene solución factible.

La prueba de estas dos propiedades se sigue directamente del hecho que $A\underline{X} <= b$ si y sólo si, $uA\underline{X} <= ub$.

Estas propiedades son útiles ya que a través de ellas la restricción substituta puede brindar información acerca de la infactibilidad, la cancelación con valor 0 o con valor 1 de alguna variable, ie, que se aplicarán las pruebas Cero y Uno a la restricción substituta (la infactibilidad se aplicará implícitamente al obtener el vector $u = (u_1,...,u_n)$.

Claramente, teniendo u_i restringido a valores no-negativos, un número infinito de restricciones substitutas pueden ser desarrolladas para el mismo problema. Además, claramente algunas de estas restricciones pueden ser superiores a las otras en el sentido de que aquéllas pueden brindar mejor información acerca de la optimalidad y la factibilidad del problema relativo a sus variables libres actuales.

La observación anterior condujo a Glover a pensar una definición de la fortaleza de una restrcción substituta. Dada tal definición, se puede desarrollar la restricción substituta más fuerte para el problema. La definición de Glover es la siguiente:

Restricción Substituta de Glover.- Supóngase que el problema está dado en forma matricial por min $\{CX \mid AX \le b,\ x_j = (0,1),\ j \text{ en } N\}$. La restricción substituta $u^1(AX - b) \le 0$ es más fuerte que la restricción substituta $u^2(AX - b) \le 0$ relativa a la actual solución parcial G_t (o a las variables libres $N - G_t$) si:

$$\min_{x} \{CX \mid u^1(AX-b) \le 0\} \ge \min_{x} \{CX \mid u^2(AX-b) \le 0\}$$

donde la minimización es sobre los valores binarios de las variables libres con las restantes variables especificadas por la solución parcial.

Una justificación intuitiva de la definición de Glover de la fortaleza de una restricción substituta es la siguiente: Una propiedad deseable en la restricción substituta es que ésta delimite mejor los valores de las variables. En cierto sentido, ésto es equivalente a proveer un espacio de soluciones más restrictivo (aunque no necesariamente en el sentido de que un espacio más restrictivo es un subconjunto propio de uno menos restrictivo). Claramente, este resultado es efectuado si el mínimo de una función objetivo dada sujeta a la restricción substituta más fuerte es mayor que el valor óptimo sujeto a la restricción substituta más débil.

Excepto para el caso donde hay sólamente dos restricciones originales, el esfuerzo computacional asociado con el problema general puede ser tedioso. Esto limita el uso práctico de la definición de Glover.

Un desarrollo posterior debido a Geoffrion (1969) muestra como puede ser modificada la definición dada por Glover de manera que la obtención de la restricción substituta puede efectuarse de manera relativamente sencilla y por lo tanto hacer más práctica la aplicación de una restricción substituta.

Restricción Substituta de Geoffrion.- Geoffrion propone que además de combinar las restricciones $AX \le b$ en la forma $uAX \le ub$ como sugirió Glover, la restricción $CX \le z^*$ o $z^* - CX$ debe ser aumentada con peso unitario, es decir, que se forma la matriz $\underline{\mathbf{A}}$ constituída por A y por la restricción $cx \le z^*$ (ie, se agrega una columna y un renglón a A) y se procede a formar la restricción substituta con esta nueva matriz $\underline{\mathbf{A}}$ en donde $u = (u_1, u_2, \ldots, u_n, 1)$. La justificación para agregar esta restricción adicional es que la resultante restricción substituta puede producir un valor objetivo CX que es mejor que el valor óptimo actual z^*.

Geoffrion define la fortaleza de su restricción de la siguiente manera:

Dados $u^1 >= 0$ y $u^2 >= 0$, la restricción substituta:

$$u^1(b-AX) + (z^* - CX) \geq 0$$

es más fuerte que:

$$u^2(b-AX) + (z^* - CX) \geq 0$$

relativa a la solución parcial G_t si:

$$\max_{x=\{0,1\}} \{u^1(b-AX) + z^* - CX \mid G_t\} \leq \max_{x=\{0,1\}} \{u^2(b-AX) + z^* - CX \mid G_t\}$$

(La justificación intuitiva de la fortaleza es la misma que la dada para la de Glover). Por lo tanto, la restricción substituta más fuerte de acuerdo a la definición de Geoffrion está dada por el siguiente problema minimax:

$$\min_{u\geq 0} \max_{x=\{0,1\}} \{u(b-AX) + z^* - CX \mid G_t\} \geq 0$$

el resultado interesante acerca de la definición de Geoffrion (el problema minimax) es que la determinación de u es equivalente a la solución de un problema de programación lineal ordinario. La equivalencia es establecida de la siguiente manera:

La parte de maximización asociada con la restricción puede ser escrita como sigue:

$$\max \left\{ \sum_{i=1}^{m} u_i\left(b_i - \sum_{j\in G_t} a_{ij}x_j^t - \sum_{j\in G_t} a_{ij}x_j\right) + \left(z^* - \sum_{j\in G_t} c_jx_j^t - \sum_{j\in G_t} c_jx_j\right) \middle| u_i \geq 0, \right.$$

$$\left. \text{para toda } \; i\in M, \quad x_j = (0,1),\; j\in G_t,\text{ y } \; x_j = x_j^t,\; j\in G_t \right\}$$

$$= \sum_{i=1}^{m} b_{it}u_i + z^* - z^t$$

$$+ \max \left\{ \sum_{j\in G_t} (-1)\left(\sum_{i=1}^{m} a_{ij}u_i + c_j\right)x_j \middle| x_j = (0,1),\; j\in G_t \right\}$$

$$= \left(\sum_{i=1}^{m} b_{it}u_i + z^* - z^t\right)$$

$$+ \max \left\{ \sum_{j\in G_t} (-1)\left(\sum_{i=1}^{m} a_{ij}u_i + c_j\right)x_j \middle| 0 \leq x_j \leq 1,\; j\in G_t \right\}$$

$$= \left(\sum_{i=1}^{m} b_{it}u_i + z^* - z^t\right)$$

$$+ \min \left\{ \sum_{j\in G_t} y_j \middle| y_j \geq 0,\; y_j \geq -\left(\sum_{i=1}^{m} a_{ij}u_i + c_j\right),\; j\in G_t \right\}$$

En la derivación anterior, en la segunda igualdad se ha reemplazado la restricción $x_j = (0,1)$, j no en G_t por $0 <= x_j <= 1$, j no en G_t ya que claramente la solución óptima a:

$$\max\left\{\sum_{j\in G_t}(-1)\left(\sum_{i=1}^{m}a_{ij}u_i+c_j\right)x_j \mid 0\le x_j\le 1,\ j\in G_t\right\}$$

asigna a la variable x_j el valor 1 si:

$$\left(\sum_{i=1}^{m}a_{ij}u_i+c_j\right)<0$$

y valor 0 en otro caso.

La tercera igualdad se sigue de la teoría de dualidad. Como resultado de ésto la expresión no-lineal en u_i y x_j se ha convertido en una expresión lineal en u_i y y_j.

El problema minimax es ahora escrito como:

$$\min\ w^t = \sum_{i=1}^{m}b_{it}u_i+\sum_{j\in G_t}y_j+(z^*-z^t)$$

s.c. a

$$-\sum_{i=1}^{m}a_{ij}u_i-y_j\le c_j$$

$$u_i,\ y_j\ge 0,\qquad i\in M,\quad j\in G_t$$

Sea w_{min}^t el valor óptimo de la función objetivo del problema. Se sigue que:

(i) Si $w_{min}^t < 0$, entonces el valor máximo que puede tomar la restricción substituta es negativo, ie que no tiene solución factible binaria (pues la restricción substituta tiene signo >= 0) lo cual implica que nuestro problema en el nodo actual no puede producir una completación mejor que la óptima actual y por lo tanto el nodo está sondeado. En general, no será necesario calcular w_{min}^t ya que el problema para determinar el vector $u = (u_1,\ldots,u_m)$ se está resolviendo por medio del algoritmo primal-simplex y por lo tanto, tan pronto se tenga un valor objetivo $w^t < 0$ tendremos este resultado.

(ii) Si $w_{min}^t >= 0$, entonces G_t no ha sido sondeado, y los valores óptimos de u_i son utilizados para producir la restricción substituta a la cual se le aplicarán las pruebas Cero y Uno.

La prueba puede ser aplicada en cualquier momento ya que inicialmente $z_{iop} = z^* = c_1 + \ldots + c_n + 1$.

Veamos ahora un ejemplo que involucra las pruebas BALAS y SURROGATE.

Sea:

$$z = (3,2,5,2,3)\,(x_1, x_2, \ldots, x_5)^T$$

sujeto a

$$\begin{pmatrix} -1 & -1 & 1 & 2 & -1 \\ -7 & 0 & 3 & -4 & -3 \\ 11 & -6 & 0 & -3 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_5 \end{pmatrix} \leq \begin{pmatrix} 1 \\ -2 \\ -1 \end{pmatrix}$$

$$x_j = 0 \text{ ó } 1, \qquad \text{para toda } j$$

Apliquemos primero la prueba BALAS, calculamos primero

$$v_j = \sum_{j \in M_j} (b_{it} - a_{ij})\,; \quad M_j = \{i \mid (b_{it} - a_{ij}) < 0,\ i = 1, \ldots, m\}$$

en este caso, $b_{it} = b_i$ pues $G_0 = \{\,\}$.

$$M_1 = \{3\} \quad \rightarrow \quad v_1 = -1 - 11 = -12$$

$$M_2 = \{2\} \quad \rightarrow \quad v_2 = -2 - 0 = -2$$

$$M_3 = \{2\} \quad \rightarrow \quad v_3 = -2 - 3 = -5$$

$$M_4 = \{1\} \quad \rightarrow \quad v_4 = 1 - 2 = -1$$

$$M_5 = \{\,\} \quad \rightarrow \quad v_5 = 0$$

como $v_5 = 0$, asignando el valor 1 a la varaible x_5 obtenemos una mejor solución entera que la óptima actual (inicialmente tenemos $z^* = (3+2+5+2+3) + 1$) por lo tanto, $G_1 = \{5\}$, $z^* = ziop = 3$. Además, G_1 ya está sondeado por lo que aplicamos la regla LIFO, lo cual produce $G_2 = \{-\underline{5}\}$ con $z^t = 0$.

Apliquemos a este nodo la prueba SURROGATE, el problema para determinar u está dado por:

$$\min\ w^2 = u_1 - 2u_2 - u_3 + y_1 + y_2 + y_3 + y_4 + (3-0)$$

$$\text{s.c. a}$$

$$\begin{aligned}
u_1 + 7u_2 - 11u_3 - y_1 \quad &\leq 3\\
u_1 \quad + 6u_3 \quad - y_2 \quad &\leq 2\\
-u_1 - 3u_2 \quad - y_3 \quad &\leq 5\\
-2u_1 + 4u_2 + 3u_3 \quad - y_4 &\leq 2\\
u_i,\ y_j &\geq 0
\end{aligned}$$

la solución óptima a este problema es:

$$w^2_{min} = -\frac{64}{65} + 3 = \frac{131}{65}$$

$$u_1 = 0$$

$$u_2 = \frac{31}{65}$$

$$u_3 = \frac{2}{65}$$

ya que $w_{min}{}^2 > 0$, G_2 no ha sido sondeado y procedemos a construir la restricción substituta. Tenemos que $x_5 = 0$ y que la restricción substituta está dada por:

$$\begin{pmatrix} 0 & \frac{31}{65} & \frac{2}{65} \end{pmatrix} \left[\begin{pmatrix} 1 \\ -2 \\ -1 \end{pmatrix} - \begin{pmatrix} -1 & -1 & 1 & 2 \\ -7 & 0 & 3 & -4 \\ 11 & -6 & 0 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \right] \geq -3 + (3\ \ 2\ \ 5\ \ 2) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

efectuando las operaciones indicadas tenemos que la restricción substituta es:

$$118x_2 + 418x_3 \leq 131$$

lo cual implica que x_3 debe tomar valor 0, dando $G_3 = \{ -\underline{5}, -\underline{3} \}$.

Hemos presentado pruebas sondeadoras con las siguientes características:

1.- Pruebas sondeadoras que conducen a una "buena" solución al inicio de la enumeración como son B - Balas y M - Ram Var Men Pen.

2.- Pruebas sondeadoras que producen cancelaciones como los son C - Ceiling, I - Infactibilidad, Z - Cero, U - Uno, P - Penalidades y S - Surrogate.

3.- Pruebas sondeadoras que hacen una selección "inteligente" de las variables a las que se les asignará algún valor binario como son B -Balas, M - Ram Var Men Pen, A - Ram Var Max Pen y E - Preferset.

4.- Pruebas sondeadoras que permitan conocer si el espacio de soluciones del P.P.L.E. (o uno semejante a el, por ejemplo el espacio de soluciones del problema lineal asociado al P.P.L.E.) es factible como lo es la prueba D - Proglin.

Consideramos que las pruebas sondeadoras presentadas son suficientes debido a que creemos que en su conjunto cubren las distintas situaciones que se pueden presentar en la resolución de un P.P.L.E.

2.4. EJEMPLO COMPLETO DE ENUMERACION IMPLICITA.

Resolvamos el siguiente Problema de Programación Entera:

$$\text{Min } z = 2x_1 + 4x_2 + 4x_3 + 8x_4 + 8x_5$$

s.c.

$$-8x_1 + 4x_2 + x_3 - x_4 - 5x_5 \leq -5$$

$$-6x_1 + 3x_2 + 2x_3 - x_5 \leq -2$$

$$2x_1 - 9x_2 - 3x_3 + 2x_4 - 3x_5 \leq -4$$

$$x_j = (0,1) \qquad j \in \overline{1,5}$$

Recordemos que antes de aplicar cada prueba se verificará si el nodo actual puede producir una mejor solución ($z^l < z^*$) ó una solución alternativa ($z^l = z^*$) y en caso afirmativo (en caso negativo el nodo esta sondeado y se aplicará la regla LIFO) se verificará si la completación trivial (asignar a todas las variables libres del nodo actual el valor 0) es factible en cuyo caso se habrá llegado a una mejor solución entera ó a una solución alternativa (el nodo estará sondeado y se aplicará la regla LIFO), en caso de que la completación trivial no sea factible se procederá a realizar la prueba seleccionada.

1.- El nodo en el cual iniciamos la enumeración es aquél que tiene a todas las variables como variables libres, ie que el nodo actual tiene a $F = \{1,2,3,4,5\}$, $L = \{\}$, $X = \{\}$, $G = \{\}$. Hagamos, inicialmente, $z^* = 2 + 4 + 4 + 8 + 8 + 1$.

2.- Apliquemos la prueba B - Balas. Procedemos a encontrar el conjunto preferido con la menor cardinalidad.

i) De la primera desigualdad tenemos que:

$$8x_1 + x_4 + 5x_5 >= 5 + 4x_2 + x_3 >= 5$$

lo cual implica que:

$$8x_1 + 5x_5 >= 5 - x_4 >= 4$$

por lo que el conjunto preferido para la primera restricción es $\{1,5\}$

ii) De la segunda desigualdad tenemos que:

$$6x_1 + x_5 >= 2 + 3x_2 + 2x_3 >= 2$$

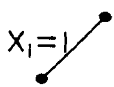
lo cual implica que:

$$6x_1 >= 2 - x_5 >= 1$$

ie, que $x_1 >= 1/6$, por lo que concluímos la prueba con el resultado de que la variable x_1 debe tomar valor 1. Obteniendo con ello:

$$G_1 = \{\underline{+1}\},\ F = \{2,3,4,5\},\ L = \{1\},\ X = \{\}$$

el árbol correspondiente es:



Notemos que hemos enumerado implícitamente la mitad de todas las combinaciones posibles.

3.- Apliquemos la prueba M - Ram Var Men Pen. Resolvamos el problema lineal (por medio del algoritmo dual-simplex para variables acotadas) correspondiente al nodo actual el cual es el siguiente:

$$\text{Min } z = 4x_2 + 4x_3 + 8x_4 + 8x_5 + 2$$

$$\text{s.c.}$$

$$4x_2 + x_3 - x_4 - 5x_5 \le 3$$

$$3x_2 + 2x_3 - x_5 \le 4$$

$$-9x_2 - 3x_3 + 2x_4 - 3x_5 \le -6$$

$$0 \le x_j \le 1 \quad j \in \overline{2,5}$$

la tabla óptima es la siguiente:

	x_1	x_2	x_3	x_4	h_1	h_2	h_3	
h_1	0	-1/3	-1/9	-19/3	1	0	4/9	1/3
h_2	0	1	2/3	-2	0	1	1/3	2
x_2	1	1/3	-2/9	1/3	0	0	-1/9	2/3
z	0	-8/3	-80/9	-20/3	0	0	-4/9	14/3

Calculemos las penalidades:

x_3 (variable no-básica con valor 0): d = 8/3 por forzarla a tomar valor 1.

x_4 (variable no-básica con valor 0): d = 80/9 por forzarla a tomar valor 1.

x_5 (variable no-básica con valor 0): d = 20/3 por forzarla a tomar valor 1.

x_2 (variable básica con valor 2/3):

$$x_2 + 1/3x_3 - 2/9x_4 + 1/3x_5 - 1/9h_3 = 2/3$$

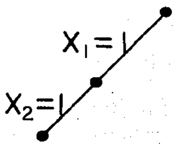
i) si forzamos $x_2 = 0$: $d_0 = -2/3(\max(-8,-20)) = 16/3$

ii) si forzamos $x_2 = 1$: $d_1 = -(2/3 - 1)(\min(40,4)) = 4/3$

por lo que asignamos a la variable x_2 el valor 1 con lo que obtenemos :

$$G_2 = \{\underline{\pm 1},+2\},\ F = \{3,4,5\},\ L = \{1,2\},\ X = \{\}$$

el árbol correspondiente es:



4.- Apliquemos la prueba E - Preferset. El subproblema en el nodo actual es:

$$\text{Min } z = 4x_3 + 8x_4 + 8x_5 + 6$$

$$\text{s.c.}$$

$$\begin{aligned} x_3 - x_4 - 5x_5 &\le -1 \\ 2x_3 \qquad - x_5 &\le 1 \\ -3x_3 + 2x_4 - 3x_5 &\le 3 \end{aligned}$$

$$x_j = (0,1) \quad j \in \overline{3,5}$$

Analizaremos únicamente la primera desigualdad ya que ésta es la única cuyo lado izquierdo es negativo.

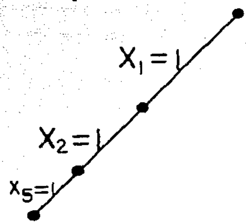
$$x_4 + 5x_5 >= 1 + x_3 >= 1$$

por lo que el conjunto mímimo preferido es $\{4,5\}$.

Elijamos para ramificar a la variable x_5, con ello obtenemos:

$$G_3 = \{\underline{\pm 1},+2,+5\},\ F = \{3,4\},\ L = \{1,2,5\},\ X = \{\}$$

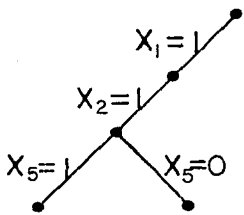
el árbol correspondiente es:



5.- Observemos que la completación trivial $x_3 = x_4 = 0$ es factible por lo que hemos llegado a nuestra primera solución factible con $z^* = 14$ y $L = \{1,2,5\}$. Este nodo está por lo tanto sondeado y aplicando la regla LIFO obtenemos:

$$G_4 = \{\underline{+1},+2,\underline{-5}\},\ F = \{3,4\},\ L = \{1,2\},\ X = \{5\}$$

el árbol correspondiente es:



6.- Apliquemos la prueba D - Proglin. El problema lineal correspondiente al nodo actual es:

$$\text{Min } z = 4x_3 + 8x_4 + 6$$

$$\text{s.c.}$$

$$x_3 - x_4 \le -1$$

$$2x_3 \le 1$$

$$-3x_3 + 2x_4 \le 3$$

$$0 \le x_j \le 1 \quad j \in \overline{3,4}$$

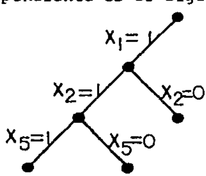
aplicando el algoritmo dual-simplex para variables acotadas llegamos a la siguiente tabla óptima:

	x_3	x_4	h_1	h_2	h_3	
x_4	-1	1	-1	0	0	1
h_2	2	0	0	1	0	1
h_3	-1	0	2	0	1	1
z	-12	0	-8	0	0	14

La solución es entera con $x_4 = 1$, $x_3 = 0$. Con el valor óptimo de la función objetivo del problema lineal asociado igual a 14 = z^*; por lo tanto hemos obtenido una solución óptima entera alternativa con L = {1,2,4}. El nodo está sondeado y aplicando la regla LIFO tenemos:

$$G_5 = \{\underline{+1}, \underline{-2}\},\ F = \{3,4,5\},\ L = \{1\},\ X = \{2\}$$

el árbol correspondiente es el siguiente:



7.- Apliquemos la prueba E - Preferset. El subproblema en el nodo actual es:

$$\text{Min } z = 4x_3 + 8x_4 + 8x_5 + 2$$

$$\text{s.c.}$$

$$\begin{aligned} x_3 - x_4 - 5x_5 &\le 3 \\ 2x_3 - x_5 &\le 4 \\ -3x_3 + 2x_4 - 3x_5 &\le -6 \end{aligned}$$

$$x_j = (0,1) \qquad j \in \overline{3,5}$$

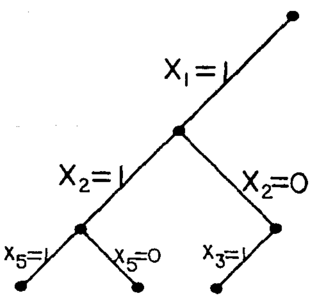
Calculemos P_i (el máximo valor que puede tomar la variable de holgura asociada a la i-ésima restricción del subproblema actual) para aquellas restricciones cuyo lado derecho es menor o igual a 0.

$$P_3 = -6 + 3 + 3 = 0$$

El mínimo de los coeficientes negativos de las variables en el tercer renglón es 3, correspondiente a las variables x_3 y x_5. Como $0 - 3 < 0$, debemos cancelar a una de estas variables con el valor 1; cancelando a la variable x_3 tenemos:

$$G_6 = \{\underline{+1}, \underline{-2}, \underline{+3}\}, \; F = \{4,5\}, \; L = \{1,3\}, \; X = \{2\}$$

el árbol correspondiente es:



8.- Apliquemos la prueba U - Uno. El subproblema en el nodo actual es:

$$\text{Min } z = 8x_4 + 8x_5 + 6$$

$$\text{s.c.}$$

$$-x_4 - 5x_5 \le 2$$

$$-x_5 \le 2$$

$$2x_4 - 3x_5 \le -3$$

$$x_j = (0,1) \qquad j \in \overline{4,5}$$

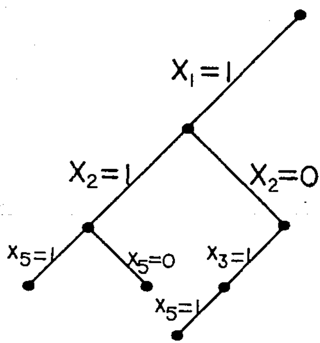
Calculemos P_i (el máximo valor que puede tomar la variable de holgura asociada a la i-ésima restricción del subproblema actual) para aquellas restricciones cuyo lado derecho es menor o igual a 0.

$$P_3 = -3 + 3 = 0$$

El mínimo de los coeficientes negativos de las variables en el tercer renglón es 3, correspondiente a la variable x_5. Como 0 - 3 < 0, debemos cancelar a esta variable con valor 1 con lo que llegamos a:

$$G_7 = \{\underline{+1}, \underline{-2}, \underline{+3}, \underline{+5}\},\ F = \{4\},\ L = \{1,3,5\},\ X = \{2\}$$

el árbol correspondiente es:



9.- Observemos que la completación trivial $x_4 = 0$ es factible con $z^l = 14$, por lo que hemos llegado a otra solución óptima alternativa con L = {1,3,5}. Este nodo está por lo tanto sondeado y aplicando la regla LIFO llegamos a que la enumeración está completa ya que todos los elementos de G_7 están conmemorados.

Las soluciones óptimas son las siguientes:

$$x_1 = x_2 = x_5 = 1, \quad x_3 = x_4 = 0$$

$$x_1 = x_2 = x_4 = 1, \quad x_3 = x_5 = 0$$

$$x_1 = x_3 = x_5 = 1, \quad x_2 = x_4 = 0$$

con $z^* = 14$.

2.5. SUGERENCIAS PARA LA APLICACION DE LAS PRUEBAS

Las siguientes observaciones y recomendaciones tienen el fin de orientar al lector en lo que se refiere a la elección de la prueba que se aplicará al nodo actual.

1.- Se recomienda que la prueba D - Proglin sea la primera prueba que se aplique al P.P.L.E. ya que si el problema lineal asociado es infactible, entonces también lo es nuestro P.P.L.E.

2.- La prueba C - Ceiling nunca producirá información acerca de las variables antes de que se obtenga la primer solución entera de nuestro P.P.L.E. debido a que el valor inicial de z^* es la suma de los coeficientes de costo más uno. La prueba calcula z^l y sólo en el caso que exista un coeficiente de costo c_j asociado a una variable libre tal que $c_j + z^l > z^*$ la prueba brindará información acerca de las variables, es claro que en todo momento de la enumeración $c_j + z^l <= c_1 + \ldots + c_n + 1$.

3.- La prueba P - Penalidades nunca producirá información acerca de las variables antes de que se obtenga la primera solución entera de nuestro P.P.L.E. debido a que en el peor de los casos $\hat{z} + d = c_1 + \ldots + c_n$ y al inicio de la enumeración z^* es la suma de los coeficientes de costo más uno por lo que nunca sucederá que $\hat{z} + d > z^*$ antes de que obtengamos la primera solución entera de nuestro P.P.L.E.

4.- Se recomienda utilizar las pruebas B - Balas y M - Ram Var Men Pen al inicio de la enumeración hasta encontrar la primera solución para nuestro P.P.L.E. debido a que estudios empíricos han demostrado que la aplicación de estas pruebas al inicio de la enumeración conducen a una "buena" solución del P.P.L.E. lo cual es de esperar que haga la enumeración bastante implícita.

5.- Se recomienda utilizar las pruebas D - Proglin y C - Ceiling (y las demás pruebas que producen cancelaciones) inmediatamente después de que se ha obtenido una mejor solución entera. Si la prueba D - Proglin trae como resultado que el nodo es infactible se aplicará la regla LIFO con lo cual es de esperar que se habrán enumerado implícitamente "muchas" asignaciones; en caso contrario (el nodo sí puede producir una mejor solución entera que la actual) la aplicación exitosa de la prueba C - Ceiling brindará información necesaria (cancelación) para la obtención de una mejor solución a nuestro P.P.L.E.

6.- Las pruebas que siempre producen cancelaciones (en el caso de que proporcionen información) son : C - Ceiling, I - Infactibilidad, Z - Cero, U - Uno, P - Penalidades y S - Surrogate. Las pruebas que producen cancelaciones tienen la ventaja de que hacen la enumeración más implícita (en caso de que brinden información) que las pruebas que producen asignaciones.

7.- Es de esperar que si la prueba S - Surrogate no brinda información para el nodo actual, entonces las pruebas I - Infactibilidad, Z - Cero y U - Uno tampoco brinden información para el nodo actual ya que la prueba S - Surrogate produce una desigualdad la cual es una combinación lineal de las restricciones del nodo actual a la cual se le aplican las pruebas I - Infactibilidad, Z - Cero y U - Uno.

8.- En caso de que las pruebas sondeadoras que producen cancelaciones no brinden información, se recomienda seleccionar la prueba E - Preferset y elegir la variable preferida con mayor coeficiente de costo asociado o seleccionar la prueba A - Ram Var Max Pen. Hacer ésto problablemente nos conducirá a cancelaciones y pasos hacia atrás.

9.- La estrategia recomendada para iniciar la enumeración es seguir las observaciones y recomendaciones en el siguiente orden: 1, 4, 5 y 8.

2.6. RELACIONES ENTRE EL PROBLEMA MODIFICADO Y EL PROBLEMA ORIGINAL.

Las variables de los problemas original y modificado (estamos en el caso en que el problema original es uno de maximización y/ó tiene al menos un coeficiente de costo negativo) están relacionadas por:

$$x_j' = 1 - x_j \quad \text{si } c_j < 0$$

$$x_j' = x_j \quad \text{si } c_j \geq 0$$

queremos saber de qué forma afecta este cambio de variables a los elementos F,L,X,G y z' en los problemas Original y Modificado.

Dotemos a los anteriores elementos de subíndice o (m) para denotar que la información contenida en ese elemento hace referencia al problema original (modificado), es suficiente analizar lo que sucede para las variables x_j con $c_j < 0$ ya que para aquellas x_j con $c_j >= 0$ no se realiza ningún cambio.

- Sea $c_j < 0$ en el problema original con x_j'' en L_m : Ya que x_j'' está en L_m si y sólo si $x_j'' = 1$ entonces $x_j = 0$ ie, x_j está en X_o.

- Sea $c_j < 0$ en el problema original con x_j'' en X_m : Ya que x_j'' está en X_m si y sólo si $x_j'' = 0$ entonces $x_j = 1$ ie, x_j está en L_o.

- Sea $c_j < 0$ en el problema original con x_j'' en F_m : Ya que x_j está en F_m (es claro que $F_o = F_m$) entonces para fines de calcular el valor de la función objetivo en el problema modificado damos a x_j'' el valor cero (podríamos decir que es una "variable libre con valor cero") entonces (por el cambio de variables) damos a x_j el valor uno para fines de cálculo de la función objetivo en el problema original ("variable libre con valor uno").

Por ejemplo:

Sea $\quad \text{Max } Z = 3x_1 + 2x_2 - 3x_3 - 8x_4$

S.C.

$$x_1 + x_2 + 3x_3 - x_4 \leq 3$$

$$-x_1 + 2x_2 + 4x_3 + x_4 \leq 7$$

$$x_i = (0,1)$$

el problema equivalente con función a minimizar y vector de costos no-negativo es:

$$\text{Min } Z = 3x_1' + 2x_2' + 3x_3 + 8x_4 - 5$$

S.C.

$$-x_1' - x_2' + 3x_3 - x_4 \leq 1$$

$$x_1' - 2x_2' + 4x_3 + x_4 \leq 6$$

$$x_i = (0,1) \quad i \in \overline{3,4}$$

$$x_i' = (0,1) \quad i \in \overline{1,2}$$

si $G_m = (\underline{\mathbf{+1}},+4,-2)$ $(F = \{3\}$ pero $x_3 = x_3'')$ entonces $G_o = (\underline{\mathbf{-1}}, +4, +2)$ con $z_m = 3 + 8 - 5 = 6$ y $z_o = - 8 + 2 - 3 = -6$.

si $G_m = (\underline{\mathbf{+2}},+3)$ $(F = \{1,4\}$ pero $x_1 = 1 - x_1'')$ entonces $G_o = (\underline{\mathbf{-2}}, +3, +1)$ con $z_m = 2 + 3 - 5 = 0$ y $z_o = - 3 + 3 = 0$.

si $G_m = (+3,+4)$ $(F = \{1,2\}$ pero $x_1 = 1 - x_1''$ y $x_2 = 1 - x_2'')$ entonces $G_o = (+3,+4,+1,+2)$ con $z_m = 3 + 8 - 5 = 6$ y $z_o = - 3 - 8 + 3 + 2 = -6$, ie que siempre tendremos que $z_o = - z_m$ para el caso en que el problema original sea de maximización. Para el caso en el que el problema original es de minimización pero éste tiene al menos un elemento $c_j < 0$ los valores de las funciones objetivo coinciden.

3 DESCRIPCION DEL PROGRAMA

El programa está diseñado para que el usuario interactúe con él por medio de menús, en los cuales se pide al usuario que seleccione una opción.

3.1. RELACIONES ENTRE LA NOTACION DEL PROGRAMA Y LA NOTACION DE LA TEORIA.

Debido a que la resolución del monitor de la computadora no permite subíndices, negritas ni subrayado, se ha tenido que modificar la notación utilizada en la exposición de la teoría, una pantalla típica (fig. 3.1) que mostrase la información general del nodo actual tendría el siguiente aspecto:

F = (2 4) L = (1)

X = (3 5) LIOP = (5)

ZIOP = 3.000

$$\text{GLOVER} = \begin{bmatrix} -5 & -3 & +1 \\ 2 & 2 & 1 \end{bmatrix}$$

Oprime cualquier tecla para continuar.

(fig 3.1)

- En donde GLOVER es la asignación parcial G del nodo actual; en donde los elementos inferiores (1 ó 2) indican si la variable está cancelada (2) o si sólo está asignada (1).

- ZIOP representa el valor óptimo entero actual, ie ZIOP = z^*.

- LIOP representa el conjunto L correspondiente a ZIOP (sólo las variables que toman valor 1).

- L es el conjunto de variables que toman valor 1.

- X es el conjunto de variables que toman valor 0.

- F es el conjunto de variables libres.

Finalmente, como última aclaración, los números en la parte superior de GLOVER indican el número de la variable (la primera, la segunda, etc.) del problema: cabe esta aclaración ya que si el usuario introdujo el problema por medio de restricciones tiene la libertad de definir los nombres de sus variables (5 letras y/ó números, siendo el primero de ellos una letra), por ejemplo si el usuario definió su primera variable como XST el programa se referirá a ésta con el número 1 en parte superior de la asignación parcial.

3.2. LIMITACIONES DEL PROGRAMA.

Las limitaciones del programa son las siguientes:

- n,m <= 10.

- los elementos de **A** y **b** son números enteros de a lo más 5 dígitos.

- los elementos de **c** son números reales con parte entera de a lo más 5 dígitos y parte decimal de a lo más 3 dígitos.

Debemos hacer notar que todas estas limitaciones se imponen únicamente por fines de presentación, ya que internamente el programa podría manejar **A** ,b y **C** con elementos reales cada uno de ellos teniendo un máximo de 11 dígitos significativos.Estando las dimensiones (n y m) limitadas sólo por la memoria de la computadora que se utilice; se ha llegado a introducir un problema con matriz **A** de dimensiones 12 x 20.

Si el usuario introduce datos incompatibles con los requerimientos impuestos o datos sin sentido, el programa mostrará las causas probables del error y pedirá al usuario que intente de nuevo.

Se tomará como nodo inicial de la enumeración aquel que tiene a todas las variables del P.P.L.E como libres.

3.3 INTRODUCCION DEL P.P.L.E.

El P.P.L.E. que se desea resolver puede ser introducido de dos formas distintas (fig. 3.2).

a) Forma matricial

b) Forma no-matricial (introduciendo las restricciones)

M - INTRODUCIR EL PROBLEMA EN FORMA MATRICIAL.
R - INTRODUCIR EL PROBLEMA INTRODUCIENDO LAS RESTRICCIONES.
K - SALIR.
¿Qué opción eliges?

En forma Matricial :	Introduciendo las restricciones :
Tu PPLE debe tener formato :	Tu PPLE debe tener formato :
Min(Max) Z = Cx s.c. Ax <= b x = (0,1)	Min(Max) Z = c1Var1 +... +cnVarn s.c. a11Var1 +... +a1nVarn <= b1 a21var1 +... +a2nVarn <= b2 .... am1Var1 +... +amnVarn <= bn Var1,...,Varn = (0,1)
* Se introducen únicamente los elementos de A, C y b.	* Donde no es necesario escribir las variables cuyo coeficiente es nulo.
* x = (x1,...,xn) donde n es el número de columnas de la ma - triz A.	* No es necesario escribir la condición de variables binarias.

(fig 3.2)

3.3.1. FORMA MATRICIAL.

Un P.P.L.E. expresado en forma matricial tiene la siguiente forma:

$$\text{Min (o max)} \quad Z = Cx$$
$$\text{s.c}$$
$$Ax \leq b$$
$$x = (0,1)$$

en donde la **matriz de coeficientes** A es una matriz de m renglones por n columnas, el **vector de requerimientos** b es un vector columna de orden m; el **vector de costos** C es un vector renglón de orden n y el **vector de variables x** es un vector columna de orden n con componentes $x_1, x_2, \ldots, x_n$.

La secuencia para introducir el P.P.L.E. es la siguiente:

- Se indica la manera de introducir el problema en forma matricial y pide al usuario que introduzca el número de renglones de A (fig. 3.3) y permite modificar este dato.

```
Introduce el número de renglones (1 ó 2 dígitos)
de A (máximo 10) y pulsa (enter).
```

```
Tu PPLE debe tener formato:                * Todos los datos se introducen
                                             en forma horizontal.
      Min(Max)  Z = Cx
                 s.c.                      * Después de introducir cada nú-
                Ax <= b                      mero oprime la tecla (enter).
                 x  = (0,1)
                                           * Tanto los elementos de A como
* Si deseas modificar los datos,             los de b deben ser enteros de
  termina de introducir todos los            a lo más 5 dígitos.
  datos y espera la opción Modi-
  ficar.                                   * Los elementos de C pueden ser
                                             números reales con parte entera
                                             de a lo más 5 dígitos y parte
                                             decimal de a lo más 3 dígitos.
```

(fig 3.3)

- Se pide al usuario introduzca el número de columnas de A y permite modificar este dato.

- Se pide al usuario que introduzca el vector C.

- Se pide al usuario introduzca los renglones (uno a la vez) de A.

- Se pide al usuario que introduzca el vector b.

3.3.2. FORMA NO-MATRICIAL.

La secuencia para introducir el P.P.L.E. es la siguiente:

- Se pide al usuario que introduzca el número de restricciones del problema y permite modificar este dato.

- Se pide se introduzca el número de variables del problema y permite modificar este dato.

- Se indica la manera de introducir el problema en forma no-matricial y pide al usuario que introduzca las restricciones (una a la vez) del problema (fig. 3.4) y permite modificar la restricción que se está introduciendo.

Introduce la 1ª restricción

- Considera como una unidad a la variable junto con su coeficiente y signo, por ejemplo: +31YY1 ó -RST (sin ningún espacio intermedio).
- Una unidad debe estar separada de otra por medio de un espacio.
- Un espacio debe preceder y continuar al signo "<=".
- Los coeficientes de las variables en las restricciones deben ser enteros de a lo más 5 dígitos.
- Las variables pueden consistir de a lo más 5 caracteres (letras y/ó números), el primero de ellos debe ser una letra.
- Los coeficientes de las variables en la función objetivo pueden ser números reales con parte entera de a lo más 5 dígitos y parte decimal de a lo más 3 dígitos.
- Oprime (enter) para finalizar cada restricción y la función objetivo
- Ejemplo de restricción válida: 3x1 +4x2 -x3 +12345x4321 <= 2.
- Si el coeficiente de la primera variable es positivo se omite "+".

(fig 3.4)

- Se pide al usuario que introduzca la función objetivo y permite modificar ésta.

3.4 EDICION DEL PROBLEMA.

El programa está provisto de opciones que permiten modificar el problema debido a que al introducir los datos del problema es fácil cometer errores.

Es necesario hacer la siguiente aclaración: El usuario podrá introducir el problema por medio de cualesquiera de las dos formas indicadas anteriormente (matricial y no-matricial), sin embargo, las modificaciones se efectuarán de forma matricial.

Se mostrarán en la pantalla C, b y A y se dará la oportunidad de modificar: C, b, A, el número de renglones de A y el número de columnas de A ó continuar con la introducción del problema (fig. 3.5).

```
  C - Modificar C   B - Modificar b   A - Modificar A   D - Continuar
  M - Modificar el número de renglones de A
  N - Modificar el número de columnas de A
** Después de cada dato deber oprimir [enter].   ¿ Qué opción eliges ?

C = ( 2.000  4.000  6.000  8.000  12.000)
b = (     1     -2     -1)

      [  -1    -1    1    2    -1 ]
A =   [  -7     0    3   -4    -3 ]
      [  11    -6    0   -3    -3 ]
```

(fig 3.5)

- Si se elige la opción C - Modificar C, se muestra el vector C y se pide al usuario que lo vuelva a introducir (deberá introducirse todo el vector), una vez hecho esto se regresará al menú de modificación (fig. 3.5).

- Si se elige la opción B - Modificar b, se muestra el vector b y se pide al usuario que lo vuelva a introducir (deberá introducir todo el vector), una vez hecho esto se regresa al menú de modificación (fig. 3.5).

- Si se elige la opción A - Modificar A, se muestra la matriz A y se dan las opciones de introducir un renglón o una columna.

  - Si se elige R - Introducir un renglón, se pide al usuario el número del renglón que desea introducir y a continuación se pide se introduzca el renglón seleccionado, una vez hecho esto se regresa al menú de modificación (fig. 3.5).

  - Si se elige C - Introducir una columna, se pide al usuario el número de la columna que desea introducir y a continuación se pide se introduzca la columna seleccionada, una vez hecho esto se regresa al menú de modificación (fig. 3.5).

- Si se elige la opción M - Modificar el número de renglones de A, se muestra A y se pide al usuario que introduzca el número de renglones que desea que tenga A y permite editar este número.

  - Si la dimensión de la matriz A aumenta se pide se introduzcan los renglones nuevos de A (uno a la vez), una vez hecho esto se muestra el vector b y se pide se introduzca éste (ya que al aumentar el número de renglones de A la dimensión de b aumenta), una vez hecho esto se regresa al menú de modificación (fig. 3.5).

  - Si la dimensión de la matriz A no aumenta, sea r el número de renglones que el usuario desea, entonces los primeros r renglones serán seleccionados para formar la nueva matriz de coeficientes, los primeros r elementos de b formaran el nuevo vector b y se regresará al menú de modificación (fig. 3.5).

- Si se elige la opción N - Modificar el número de columnas de A, se muestra A y se pide al usuario introducir el número de columnas que desea que tenga A y permite editar este número.

- Si la dimensión de la matriz A aumenta se pide se introduzcan las columnas nuevas de A (una a la vez) (para el caso no-matricial se pedirán también las nuevas variables), una vez hecho esto se muestra el vector C y se pide se introduzca éste (ya que al aumentar el número de columnas de A la dimensión de C aumenta), una vez hecho esto se regresa al menú de modificación (fig. 3.5).

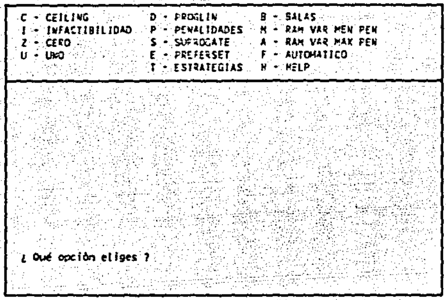
- Si la dimensión de la matriz A no aumenta, sea c el número de columnas que el usuario desea, entonces, las primeras c columnas serán seleccionadas para formar la nueva matriz de coeficientes, los primeros c elementos de C formarán el nuevo vector C y se regresará al menú de modificación (fig. 3.5).

- Si se elige la opción D - Continuar, se muestran C, b y A y se pregunta al usuario si el P.P.L.E. que desea resolver es uno de minimización, en caso negativo se supone que el problema es uno de maximización.

  - En el caso en que el P.P.L.E. sea uno de maximización y/ó exista al menos un coeficiente de costo negativo, se informará al usuario que el programa trabaja sólo con P.P.L.E. en forma de minimización con C >= 0 y muestra en la pantalla C', b' y A' que conforman un nuevo P.P.L.E. (**problema modificado**) equivalente al original (**problema original**) con C' >= 0 y función objetivo a minimizar, muestra también el cambio de variables realizado para transformar el problema original en el modificado.

    Para comodidad del usuario el programa escribirá x_j en vez de x_j' para aquellas variables con $c_j >= 0$.

    Una vez que se oprime cualquier tecla para seguir, se muestra la nueva función objetivo a minimizar (junto con el cambio de variables) y una vez que se decide seguir se muestran las distintas pruebas que se pueden aplicar al problema, además de mostrar dos opciones de ayuda (T -Estrategias), (H - Help) y una opción para salir del programa (K - Salir) (fig. 3.6).



(fig 3.6)

- El elegir la opción F - Automático permite visualizar la solución óptima (si ésta existe) del problema, sin que el usuario intervenga en la búsqueda de ésta; la prueba se puede seleccionar en cualquier momento de la enumeración.

- Si se elige la opción T - Estrategias aparecerá en la pantalla información de cómo realizar la búsqueda de la solución óptima; aclaremos que estas observaciones tienen una justificación empírica ya que el encontrar la solución óptima de manera "rápida" depende principalmente de los elementos de la matriz de coeficientes A y del vector b.

- Si se elige la opción H - Help (ayuda) aparecerá en la pantalla información general acerca del programa, así como una descripción de cada prueba.

- Si se elige la opción K - Salir, se permite al usuario modificar el problema actual o introducir un nuevo problema o salir definitivamente el programa.

  - Si se elige la opción M - Modificar el Problema Actual, el programa mostrará el menú de modificación (fig. 3.5).

- Si se elige la opción N - Introducir un Nuevo Problema, se mostrará la pantalla que permite la elección de introducir el problema en forma matricial ó introduciendo las restricciones (fig. 3.2).

- La elección de K - Salir, permite salir del programa definitivamente, aparecerá el indicador del DOS.

- Si se elige aplicar una prueba al problema, (opciones C, I, Z, U, D, P, S, E, B, M, A) se aplicará la prueba elegida y se mostrará su resultado.

- Si el problema es uno de minimización con C >= 0 se darán las opciones de continuar con la aplicación de las pruebas y la de visualizar información de caracter general para el problema (F, L, X, LIOP, ZIOP, GLOVER).

- Si el problema es uno de maximización y/ó con al menos un coeficiente de costo negativo se darán las opciones de continuar con la aplicación de las pruebas, la de visualizar información de caracter general (F, L, X, LIOP, ZIOP, GLOVER) para el problema original y la opción de visualizar información de caracter general (F, L, X, LIOP, ZIOP, GLOVER) para el problema modificado.

Una vez resuelto el problema, se muestran en la pantalla las variables que toman valor 1 (las demás toman valor 0), así como el valor óptimo.

En el caso en que se tengan problemas Original y Modificado, la solución mostrada corresponderá al problema modificado y una vez que se desee seguir se preguntará al usuario si desea ver la solución óptima para el problema original, en caso afirmativo se mostrarán las variables que toman valor 1 además del valor óptimo. Después de visualizar la solución óptima (si ésta existe) se permitirá modificar el problema actual, introducir uno nuevo o salir definitivamente del programa.

CONCLUSIONES.

Podemos concluir que la eficiencia de la enumeración implícita depende directamente de lo siguiente:

1.- De que tan bien sean diseñadas las pruebas sondeadoras, las cuales tienen la finalidad de identificar tantas soluciones no-prometedoras como sea posible para descartarlas automáticamente, así como de elegir "inteligentemente" a las variables a las que se les asignará algún valor binario.

2.- De la estrategia que se siga en la enumeración. Consideramos que una buena estrategia es aquella que logra obtener una "buena" solución (óptima ó cercana a la óptima) al inicio de la enumeración y que una vez que se obtiene ésta aplica a los nodos restantes por enumerar pruebas que produzcan cancelaciones.

3.- De la experiencia que se tenga en la resolución de P.P.L.E. con el enfoque de la enumeración implícita.

La principal ventaja del enfoque de la enumeración implícita es que al aplicar ésta no se "acarrean" los errores de redondeo, lo cual hace que éstos sean prácticamente inexistentes (en los algoritmos de cortaduras ésto representa una severa dificultad).

Consideramos que un sistema de enumeración implícita debe proveer al usuario de pruebas sondeadoras con las siguientes características:

1.- Pruebas sondeadoras que conduzcan a una "buena" solución al inicio de la enumeración como son B - Balas y M - Ram Var Men Pen.

2.- Pruebas sondeadoras que produzcan cancelaciones como los son C - Ceiling, I - Infactibilidad, Z - Cero, U - Uno, P - Penalidades y S - Surrogate.

3.- Pruebas sondeadoras que hagan una selección "inteligente" de las variables a las que se les asignará algún valor binario como son B -Balas, M - Ram Var Men Pen, A - Ram Var Max Pen y E - Preferset.

4.- Pruebas sondeadoras que permitan conocer si el espacio de soluciones del P.P.L.E. (o uno semejante a el, por ejemplo el espacio de soluciones del problema lineal asociado al P.P.L.E.) es factible como lo es la prueba D - Proglin.

BIBLIOGRAFIA

E. B. Koffman, Problem Solving and Structured Programming in PASCAL, Addison-Wesley, Massachusetts, 1981.

K. G. Murty, Linear and Combinatorial Programmiq, John Wiley & Sons, New York, 1976.

H. Salkin, Integer Programming, Addison - Wesley, Reading, Massachusettes, 1975.

M. Simonnard, Linear Programming, traducido del Frances por W. S. Jewell, Prentice Hall, Englewood Cliffs, N.J., 1966.

H. TAHA, Integer Programming : Theory, Applications and Computations, Academic Press, New York, 1975.

M. Yester, Using Turbo Pascal, Que Corporation, 1989.