9
2ej

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"ACATLAN"

PROGRAMACION LINEAL: UN ENFOQUE GERENCIAL

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE

LICENCIADO EN ACTUARIA

P R E S E N T A

ENRIQUE MARTIN MURILLO OTHON

ASESOR

M. LUZ MARIA RANGEL NAFAILE

ACATLAN, MEX. 1987

I N D I C E

INTRODUCCION

Es un hecho innegable que entre las herramientas matemáticas para toma de decisiones disponibles actualmente, la de uso mas extendido es la Programación Lineal, o PL. Sin embargo, existen aun muchos directivos, tanto en el sector público como en el privado, que no estan familiarizados con sus nociones operativas, posibles aplicaciones y reconocidas limitaciones. Es primeramente a ellos a quienes está dirigido este trabajo. Su objetivo es transmitir un conocimiento introductorio de carácter práctico sobre el uso y ventajas de esta técnica.

El recurso mas escaso de todo director es su tiempo. Por eso juzgué que la utilidad de este trabajo dependería mucho de su concisión. Un gerente difícilmente podrá contar con el número de horas requerido para adentrarse en alguno de los numerosos tratados de PL actualmente editados. Además, la mayoría de ellos se limitan a una exposición teórica complementada con algunos ejemplos también teóricos. Sin duda alguna, las matemáticas empleadas resultan fascinantes para algunos lectores, pero frecuentemente lo que un gerente atareado desea es evaluar la utilidad de la técnica para *su problema concreto* y --si se despierta su interés--, aprender a utilizarla en forma práctica.

Para responder a esta necesidad he constituido este trabajo de tesis con dos elementos que se complementan: un paquete computacional interactivo y el presente manual introductorio. El resultado final es un material didáctico que permite, a personas de muy diversa preparación profesional, aprender a analizar problemas concretos de optimización con esta técnica.

La idea original para un producto de esta naturaleza la debo al Ing. Carlos Alejandro Armenta Pico, profesor del área de Análisis de Decisiones en el Instituto Panamericano de Alta Dirección de Empresa (I.P.A.D.E.). Esta institución tiene un alumnado muy heterogéneo; el reto consistía en desarrollar un paquete interactivo que pudiera utilizar, sin previa instrucción, cualquier persona que poseyera un conocimiento elemental de PL. Después de trabajar unos meses en el proyecto, decidí que sería conveniente escribir un breve texto para proporcionar también este conocimiento.

El manual que el lector tiene ahora en sus manos se divide en dos partes. En la primera se expone, en dos capítulos, el material estrictamente necesario para iniciar el estudio de la técnica. La segunda parte, dedica tres capítulos al análisis de algunos problemas típicos que pueden ser atacados mediante PL. Estos ejemplos tienen un grado creciente de complejidad y su fin es reforzar y ejercitar el material previamente expuesto.

El conocimiento que deseo transmitir no incluye los detalles matemáticos del Método Simplex, el procedimiento estandar para la resolución de modelos lineales. Inicialmente, un director no tiene ni el tiempo ni la necesidad de profundizar tanto. Mas bien he preferido mostrar la profundidad del análisis que se puede hacer de muchos sistemas de la vida real utilizando PL. El texto puede leerse de corrido con sólo tener una familiaridad con los sistemas de ecuaciones y de desigualdades lineales.

El programa, bautizado con el nombre de Paquete Interactivo de Programación Lineal (P.I.P.L.), se utilizó para generar los listados de resultados que aparecen en el manual. He incluido un resumen de las características técnicas y el listado completo del paquete en el Apéndice II.

Estimo que una primera lectura de este trabajo puede hacerse cómodamente en una o dos horas, y es suficiente para decidir si el tema merece un estudio mas profundo.

Primera Parte : Fundamentos de Programacion Lineal.

Esta sección es una breve introducción a la técnica y arte de analizar sistemas de la vida real mediante modelos lineales. En el primer capítulo se presenta el Caso Dulcería López, un sencillo problema de asignación de recursos, y se muestra como construir un modelo lineal que lo representa adecuadamente. Esto permite además introducir la terminología propia de la Programación Lineal. La laboriosa tarea de resolver el modelo es normalmente confiada a una computadora, pero aún así es importante conocer, aunque sea a grandes rasgos, los métodos de solución. Sólo así pueden explicarse ciertas situaciones especiales que pueden presentarse en la solución óptima, y el importante tema del Análisis de Sensibilidad. Estas cuestiones se tratan en el capítulo segundo, donde podrá apreciarse cuánta información relevante encierra el problema, aparentemente trivial, del Sr. López.

Capítulo 1

Construcción de Modelos Lineales

Uno de los problemas que con mayor frecuencia debe enfrentar un director, un gerente, o cualquier otro profesional que debe tomar decisiones, es el de asignar de manera óptima un conjunto de recursos limitados o escasos a la ejecución de dos o más actividades productivas que compiten por el uso de esos recursos. En concreto, considérese el siguiente ejemplo:

Caso 1: Dulcería López

Ernesto López es propietario de una pequeña dulcería. Desde hace tiempo ha obtenido buenos resultados vendiendo chocolates que él mismo produce en pequeñas cantidades.

El Sr. López elabora dos variedades de chocolates: "Dados" y "Dados Rellenos". Ambos son pequeños bloques de chocolate macizo en forma de dado; los segundos tienen además un relleno de crema batida que los hace particularmente apreciados por su clientela.

El Sr. López, ayudado por su esposa, lleva a cabo su pequeña operación de producción el domingo por la tarde a fin de tener los chocolates en su mostrador el lúnes por la mañana. Generalmente estos han desaparecido totalmente para el día siguiente, pues la dulcería está situada en una zona escolar. Este éxito le ha hecho pensar en aumentar su producción, pero ello le ha llevado a considerar que debe hacer un uso más eficiente de sus recursos.

Para la elaboración de los chocolates se requieren tres productos principales: chocolate amargo de mesa, jarabe de chocolate y crema batida. Los primeros dos se obtienen fácilmente en cualquier mercado. En concreto el Sr. López se ha puesto de acuerdo con un amigo que tiene un negocio de abarrotes y que le mandará semanalmente una caja de 24 kg. de chocolate de mesa y dos latas de 5 kg. de jarabe de chocolate.

La crema batida, sin embargo, representa un problema aparte, pues es de una marca extranjera que no se consigue en el país. Es precisamente la calidad de este producto la responsable de la gran aceptación de los "Dados Rellenos", y el Sr. López no piensa dejar de utilizarla. El obtiene la crema a través de su hija, que es aeromoza, y que gustosamente le trae un envase de 500 gramos cada semana. Ella se ha ofrecido a traerle hasta tres envases semanalmente si él así lo desea.

Antes de tomar ninguna decisión, siendo un hombre ordenado, López pone por escrito un resumen (Tabla 1-1) de los datos relevantes de su problema.

El Sr. López desea obtener la máxima utilidad posible y debe decidir la cantidad a producir de cada variedad de chocoloate.

TABLA 1-1 : USO DE RECURSOS EN LA PRODUCCION DE CHOCOLATE

Para producir 1 kg. de	Chocolate de Mesa	Jarabe de Chocolate	Crema Batida	Utilidad
Dados	0.8 kg.	0.2 kg.	-	$ 1,000.
Dados Rellenos	0.4 kg.	0.5 kg.	0.1 kg.	$ 1,500.
Disponibilidad semanal	24 kg.	10 kg.	1.5 kg.	

o

Este caso es un ejemplo típico de la clase de problemas de decisión que pueden resolverse mediante la PL. El proceso de solución pasará por tres etapas:

a) Construir un modelo matemático que sirva para "representar" en el papel los procesos reales del sistema.

b) Resolver en una computadora el modelo matemático.

c) Aplicar la solución obtenida en el papel al sistema real. Si el modelo matemático es una buena "aproximación" del sistema real, entonces la solución del modelo será, a su vez, una buena "aproximación" de la solución óptima del sistema real.

En lo que resta del capítulo se mostrará cómo implementar la primera etapa al problema del Sr. López.

La construcción de un modelo de Programación Lineal para representar un sistema real puede subdividirse en tres pasos. Primeramente, debe especificarse cuáles son las incógnitas del problema; esto es, qué es lo que se desea averiguar; cuál es la decisión que se debe tomar. En el caso del Sr. López la decisión es qué cantidad debe producir de "Dados" y qué cantidad de "Dados Rellenos" para obtener la máxima utilidad posible con los recursos de que dispone. Para expresar esto en términos matemáticos se representará mediante una variable la cantidad de chocolate de cada tipo a producir. Por ejemplo, se puede escribir:

Sea x = Cantidad en Kg. de "Dados" a producir, y

y = Cantidad en Kg. de "Dados Rellenos" a producir.

En otras palabras, x y y son las incógnitas de nuestro problema. Por lo tanto, una vez que se resuelva el modelo, el valor que tomen las variables x y y será el que indique la cantidad que se debe producir de cada tipo de chocolate para obtener la utilidad máxima deseada. Es por eso que a las variables x y y se les denomina *Variables de Decisión*.

En segundo lugar se debe expresar el objetivo del problema en términos de las variables de decisión. En particular el objetivo del Sr. López es obtener la máxima utilidad posible de la venta de chocolates. La Tabla 1-1 indica que cada Kg. producido de "Dados" genera $ 1,000 de utilidad y cada Kg. de "Dados Rellenos", $ 1,500 de utilidad (pues se venden todos los chocolates producidos). Utilizando las variables ya definidas se puede escribir:

$$\text{Utilidad percibida} = 1,000\,x + 1,500\,y\,.$$

Si se denota --siguiendo una práctica habitual en PL-- por "z" a la utilidad percibida, y se tiene presente que el objetivo es maximizar esta variable, la expresión puede quedar de la siguiente manera:

$$\text{Maximizar} \quad z = 1,000\,x + 1,500\,y\,.$$

Esta ecuación expresa en forma matemática el objetivo del Sr. López (maximizar la utilidad percibida de la producción de "Dados" y "Dados Rellenos"). De ahí que reciba el nombre de *Función Objetivo*. Las constantes 1,000 y 1,500 se denominan *Contribuciones* o *Costos* de las respectivas variables en la función objetivo.

Por último, hay que considerar cuáles son las *Restricciones* que limitan el crecimiento de la función objetivo. En este caso la utilidad se ve restringida por la cantidad de materia prima disponible. Por ejemplo, el consumo semanal de chocolate de mesa no puede exceder de 24 kg., cantidad de que actualmente dispone el Sr. López. Estas restricciones físicas también deben ser expresadas en términos matemáticos.

Se sabe que el consumo de chocolate de mesa depende de la cantidad producida de "Dados" y de "Dados Rellenos". Por tanto se utilizarán nuevamente las variables de decisión x y y que representan precisamente estas cantidades. Ahora bien, por cada kilogramo de "Dados" que se elabore, se consumirán 0.8 Kg. de chocolate de mesa, mientras que la misma cantidad de "Dados Rellenos" requiere 0.4 Kg. Por lo tanto, se puede escribir:

$$\boxed{\text{Cantidad de chocolate de mesa consumida (en kg.)}} = 0.8\,x + 0.4\,y\,.$$

Ahora hay que indicar que el consumo de chocolate no puede exceder de 24 Kg. Esto se señala con una desigualdad "≤"; de modo que la primera restricción del problema toma la siguiente forma:

$$0.8\,x + 0.4\,y \leq 24\,.$$

Análogamente se pueden construir las restricciones que corresponden a los otros dos ingredientes, el jarabe de chocolate y la crema batida. Estas quedarían de la siguiente manera:

$$0.2\,x + 0.5\,y \leq 10 \qquad \text{Jarabe}$$

$$0.1\,y \leq 1.5 \qquad \text{Crema}\,.$$

Nótese que en la última restricción la variable x no ha sido incluida. Esto simplemente refleja el hecho de que los "Dados" no requieren crema batida para su elaboración, o sea que su consumo de crema es de 0. Las constantes 24, 10 y 1.5 (que, en este modelo, representan los recursos disponibles) se denominan *Lados Derechos de las Restricciones*, o también, *Disponibilidades*.

Sólo resta una cosa por agregar al modelo, pero es muy importante. Por causa de los métodos de solución que se utilizan, las variables en un modelo de PL no deben asumir nunca valores negativos. Esto no representa ningún inconveniente, pues de todas formas sería imposible hablar, por ejemplo, de producir -8 kg de "Dados Rellenos". Por lo tanto, hay que considerar dos restricciones adicionales, que son:

$$x \geq 0$$

$$y \geq 0 \; .$$

Estas se conocen como *Restricciones de No Negatividad* y normalmente se escriben al final del modelo.

El modelo completo para el problema del Sr. López queda expresado de la siguiente manera:

Modelo 1 : Dulcería López

Determinar la cantidad de "Dados" y "Dados Rellenos", x y y, que debe producirse para

$$\text{Max } z = 1,000\,x + 1,500\,y \quad \text{Utilidad}$$

sujeta a las restricciones

$$
\begin{array}{rcrcll}
0.8\,x & + & 0.4\,y & \leq & 24 & \text{Chocolate} \\
0.2\,x & + & 0.5\,y & \leq & 10 & \text{Jarabe} \\
 & & 0.1\,y & \leq & 1.5 & \text{Crema} \\
 & & x,\ y & \geq & 0 & \text{No negatividad}
\end{array}
$$

El formato con el que está escrito el Modelo 1, se distingue por las siguientes características:

1) Todas las variables se encuentran en el lado izquierdo de las restricciones.
2) En el lado derecho de cada restricción hay una constante no negativa.
3) Todas las variables están restringidas a valores no negativos.

En general, los paquetes comerciales de PL están diseñados para aceptar modelos lineales expresados de esta manera.

Capítulo II

Métodos de Solución

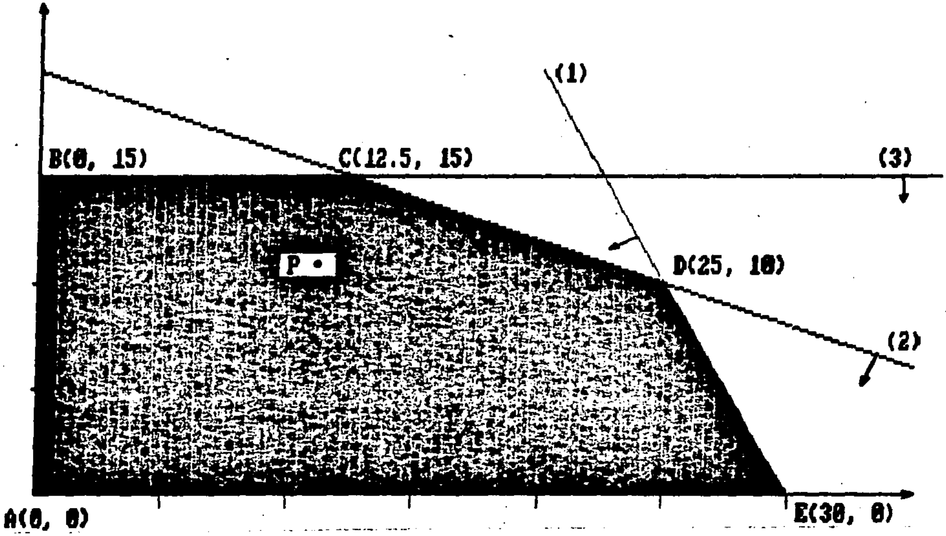
1. Un procedimiento geométrico.

En este capítulo se resuelve el Modelo 1 mediante un procedimiento gráfico que tiene la ventaja de ilustrar admirablemente el proceso de solución de un problema de programación lineal y prepara el camino para entender el método simplex que es el procedimiento estándar para atacar este tipo de problemas.

Como su nombre lo indica, el método consiste esencialmente en graficar las restricciones y la función objetivo del problema. Hecho esto es posible, por simple inspección visual y realizando algunos cálculos auxiliares, indicar cuál es la solución óptima del problema.

El modelo se reproduce en la página siguiente. Obsérvese que se han numerado las restricciones, incluso las de no negatividad. Esto permite identificarlas claramente en la gráfica 1.

GRAFICA 1



$$\text{Max } z = 1,000\,x + 1,500\,y \quad \text{Utilidad}$$

sujeta a

$0.8\,x + 0.4\,y \leq 24$	Chocolate	(1)
$0.2\,x + 0.5\,y \leq 10$	Jarabe	(2)
$0.1\,y \leq 1.5$	Crema	(3)
$x \geq 0$	No neg.	(4)
$y \geq 0$	No neg.	(5)

La región sombreada es aquella en la cual se satisfacen simultáneamente las tres restricciones, lo cual se verifica fácilmente tomando cualquier punto, ya sea dentro de la región o sobre sus contornos, y sustituyendo los correspondientes valores de x y y en el sistema de restricciones. Esta región recibe el nombre de región de soluciones factibles ya que dentro de ella se encuentran todas las soluciones posibles del problema. De ellas interesa identificar aquella que produzca la mayor utilidad.

Ahora bien, dentro de la región de soluciones factibles existe un número infinito de puntos, y cada uno de ellos representa una posible solución al modelo lineal. Encontrar la solución óptima probando el valor de la función objetivo en cada uno de estos puntos es imposible. Sin embargo, una poca de reflexión puede hacer la búsqueda bastante sencilla. En primer lugar, ningún punto interior de la región, como por ejemplo el punto P, puede ser el óptimo, puesto que se puede aumentar la utilidad moviédose hacia la derecha (esto es, aumentando la producción de "Dados") y hacia arriba (aumentando la producción de "Dados Rellenos") hasta topar con una de las fronteras de la región. Análogamente, una solución que se encuentre en un borde de la región puede ser mejorada moviéndose a lo largo del borde hasta alguna esquina o vértice. Se deduce, por tanto, que la solución óptima del problema debe encontrarse en alguno de los vértices de la región de soluciones factibles.

Para un problema pequeño como este, se podría simplemente probar los cinco vértices de la región y elegir aquel que reportara la máxima utilidad. Empero, siguiendo un procedimiento un poco más largo pero mucho más ilustrativo, se comprenderá mejor el principio operativo del método simplex.

Tomando como punto de partida el vértice A(0, 0), que lógicamente tiene un valor de utilidad $z = \$0$, se busca alguna dirección hacia donde moverse (a lo largo de un borde) que produzca un incremento en el valor de la función objetivo. Nótese que tanto el movimiento hacia arriba (lo que implica aumentar el valor de y), como hacia la derecha (aumentando el valor de x), produciría un incremento en z Si se decide (arbitrariamente) aumentar primero el valor de y, el movimiento sería hacia arriba, sobre la recta AB, hasta el siguiente vértice (no tiene caso detenerse antes). En el punto B(0, 15), el valor de la función objetivo ha aumentado a $z = 1,000 \times 0 + 1,500 \times 15 = \$22,500$, lo que representa una mejora notable. Nuevamente, se busca una dirección hacia donde desplazarse para aumentar el valor de la función objetivo. El movimiento hacia la derecha a lo largo de la recta $y = 15$, aumenta el valor de x sin disminuir el de y, lo cual causa un incremento en la función objetivo. Conviene, por tanto, moverse hacia la derecha hasta el vértice C(12.5, 15), donde $z = 1,000 \times 12.5 + 1,500 \times 15 = \$35,000$. Una vez más, se busca una dirección hacia donde viajar, desde este vértice, que sea beneficiosa para el valor de z La única posibilidad es a lo largo de la recta CD, ya que sería inútil regresar hacia B. El problema es que avanzar por la recta CD aumenta x sólo a costa de disminuir y. Se percibe claramente la escasez de los recursos, que obliga a producir menos "Dados Rellenos" a cambio de más "Dados". Concretamente, por cada unidad que x crezca a lo largo de la recta CD, y disminuirá 2/5 de unidad (ya que la pendiente de CD es -2/5). Considerando, empero, los coeficientes de la función objetivo, se observa que un aumento unitario de x incrementa la utilidad en $1,000, mientras que la correspondiente disminución en y da lugar a una pérdida de 2/5 × 1,500 = $600. Esto es, se ganan $400 netos por cada unidad adicional de x producida. Por lo tanto, el movimiento a lo largo de la recta CD hasta el punto D(25, 10) es beneficioso y, realizándolo, la función objetivo aumenta a z= $40,000. El siguiente posible movimiento sería hacia el vértice E, siguiendo la recta DE. Sin embargo, esto costaría 2 unidades de y por cada unidad adicional de x (la pendiente de DE es -2). Esto es, se perderían $3,000 por cada $1,000 ganados; evidentemente el desplazamiento hacia E no conviene. Por consiguiente, puesto que un movimiento a partir del punto D, en cualquier dirección, sólo reduciría el valor de la función objetivo, este vértice representa la solución buscada.

Solución Optima del Modelo 1	
Producir:	25 Kg. de "Dados"
	10 Kg. de "Dados Rellenos"
Utilidad máxima:	$ 40,000.00 .

2. Casos especiales en la solución de modelos lineales.

En la solución de un modelo lineal pueden darse tres situaciones especiales, que se exponen a continuación.

El modelo no tiene solución factible.

Este caso se presenta cuando el conjunto de restricciones es tal que no existe una solución al modelo que satisfaga simultáneamente todas y cada una de las restricciones.

Esto puede verse claramente haciendo una modificación al Modelo 1. Supóngase, por un momento, que el Sr. López espera recibir esta semana un pedido de chocolates surtidos por parte de un restaurante, y desea tener al menos 40 Kg. disponibles de chocolates. Esta situación quedaría representada con un modelo como el que sigue:

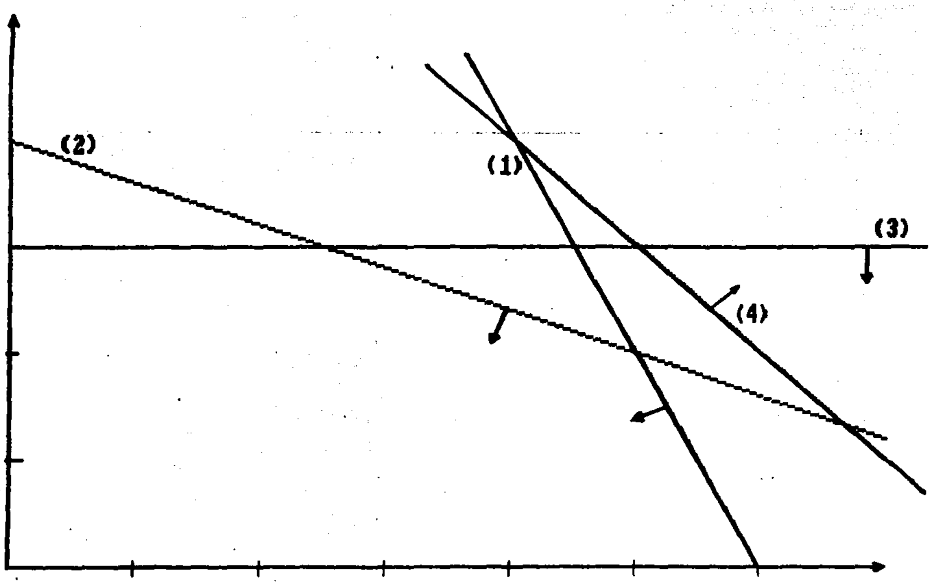
$$\text{Max } z = 1{,}000\,x + 1{,}500\,y \qquad \text{Utilidad}$$

sujeta a

$$\begin{array}{rcrcll}
0.8\,x & + & 0.4\,y & \leq & 24 & \text{(1) Chocolate} \\
0.2\,x & + & 0.5\,y & \leq & 10 & \text{(2) Jarabe} \\
 & & 0.1\,y & \leq & 1.5 & \text{(3) Crema} \\
x & + & y & \geq & 40 & \text{(4) Producción mínima} \\
 & & x,\ y & \geq & 0 & \text{No negatividad}
\end{array}$$

Se ha añadido al Modelo 1 una nueva restricción, la número (4), que determina un límite inferior de producción. La Gráfica 2 muestra claramente la incongruencia de este modelo. Nótese que no existe una región de soluciones factibles, ya que ningún punto de la gráfica puede satisfacer simultáneamente las cuatro restricciones. Esto quiere decir que con los recursos disponibles resulta físicamente imposible producir 40 Kg. de chocolates (lo cual es lógico, pues toda la materia prima suma apenas 35.5 Kg.).

GRAFICA 2



En un modelo tan pequeño como este, no resulta difícil detectar una inconsistencia en el sistema de restricciones, pero cuando se maneja un gran número de variables y restricciones, es necesaria la información que proporciona la computadora para detectar el problema.

El modelo tiene múltiples soluciones óptimas.

Ahora, volviendo al Modelo 1 original, supóngase que el Sr. López ha decidido subir el precio de los "Dados Rellenos" de tal manera que ahora la utilidad que percibe es de $2,500 por Kg. Es evidente que si sus clientes aceptan el aumento de precio, la utilidad de López aumentará. Sin embargo, cabe preguntarse si el plan de producción óptimo (25 Kg. de "Dados", 10 Kg. de "Dados Rellenos") debe cambiar con el nuevo precio. El modelo ahora es:

$$\text{Max } z = 1,000\,x + 2,500\,y \qquad \text{Utilidad}$$

sujeta a

$$\begin{aligned} 0.8\,x + 0.4\,y &\le 24 && (1) \text{ Chocolate} \\ 0.2\,x + 0.5\,y &\le 10 && (2) \text{ Jarabe} \\ 0.1\,y &\le 1.5 && (3) \text{ Crema} \\ x,\ y &\ge 0 && \text{No negatividad} \end{aligned}$$

Como no ha cambiado el sistema de restricciones, sino solamente uno de los coeficientes de la función objetivo, se puede utilizar de nuevo la gráfica 1. Partiendo del antiguo punto óptimo D, se busca alguna dirección hacia donde moverse para aumentar el valor de la función objetivo que, merced al aumento en la contribución de y, de antemano ha incrementado su valor hasta $z = 1,000 \times 25 + 2,500 \times 10 = \$50,000.00$.

En primer lugar, nótese que moverse hacia el punto E, siguiendo la recta DE, resulta ahora aún más perjudicial que antes. Cada Kg. adicional de "Dados" se produce a costa de 2 Kg. de "Dados Rellenos"; en términos monetarios, se ganarían $1,000 pero se perderían $5,000; o sea una pérdida neta de $4,000.

Sin embargo, existe la posibilidad de desplazarse hacia el punto C. Al hacer esto, disminuye la cantidad producida de "Dados" y aumenta la de "Dados Rellenos" lo que resulta atractivo debido a la diferencia de precios. Concretamente, por cada kilogramo que se deja de producir de "Dados", se ganan 2/5 Kg. de "Dados Rellenos" pues la pendiente de CD es de -2/5. Ahora bien, por el Kg. de "Dados" no producido se reduce la utilidad en $1,000, mientras que por 2/5 Kg. de "Dados Rellenos", aumenta la utilidad en $2/5 \times 2,500 = \$1,000$. Por lo tanto, a lo largo de la recta CD, la utilidad permanece constante, pues las ganancias y las perdidas ocasionadas por este movimiento se cancelan exactamente. En otras palabras, cualquier punto sobre la recta CD, incluyendo los vértices C y D, representa una solución factible al modelo que reporta la misma utilidad máxima de $50,000.

Existirían, por tanto, múltiples soluciones óptimas. Sin embargo, es importante recalcar que en todo problema de PL, si existe solución óptima, entonces por lo menos uno de los vértices de la región de soluciones factibles será solución óptima del modelo. De hecho, este es el principio de operación del Método Simplex.

El modelo tiene solución no acotada.

Considérese el siguiente modelo lineal:

$$\text{Max } z = x + 2y$$

sujeta a

$$-x + y \leq 4 \qquad (1)$$

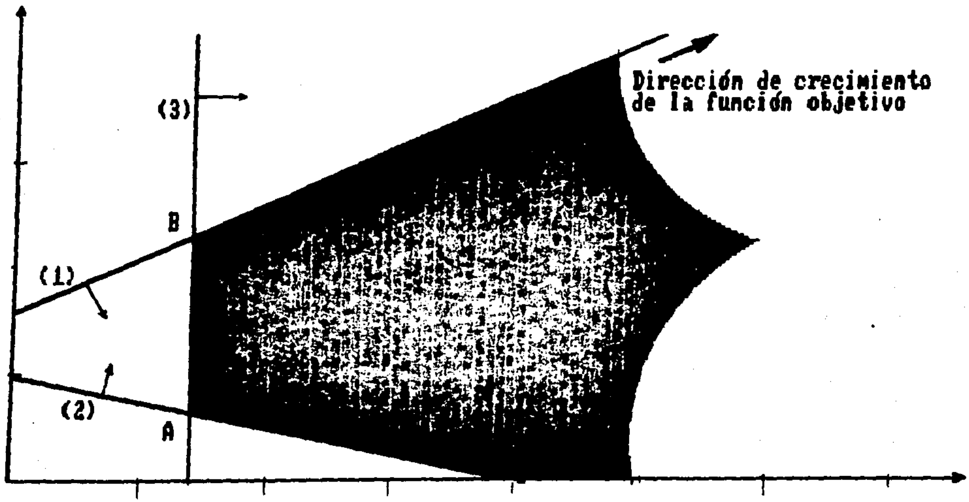
$$x + 8y \geq 16 \qquad (2)$$

$$x \geq 5 \qquad (3)$$

$$x,\ y \geq 0$$

La gráfica correspondiente aparece en la siguiente página. Obsérvese que la región de soluciones factibles está abierta por uno de sus lados. Utilizando el método geométrico, se podría por ejemplo empezar en el punto A y de ahí avanzar hacia B lo cual claramente es beneficioso. Pero hay un problema en este vértice. La única dirección de movimiento, que mejora el valor de la función objetivo, es a lo largo del borde superior de la región de soluciones factibles; esto hace que tanto x como y aumenten.

GRAFICA 3



Sin embargo, en este borde no hay un vértice donde detenerse, sino que se puede continuar indefinidamente, a la vez que aumenta sin límite la función objetivo.

Esta situación sólo puede ser consecuencia de un modelo mal formulado o incompleto. Resulta incongruente hablar, por ejemplo, de una utilidad --o cualquier otra variable del mundo real-- que puede aumentar indefinidamente. La computadora enviará un mensaje cuando encuentre este tipo de error.

3. El Método Simplex y P.I.P.L.

Hasta ahora se ha hecho el análisis del modelo lineal con un enfoque exclusivamente geométrico, logrando de esta manera introducir gran parte de la terminología propia de la PL. Ahora hay que aclarar que el método gráfico únicamente resulta aplicable a problemas pequeños, ya que para representar el modelo en un sistema de ejes coordenados se requiere de tantas dimensiones como variables tenga el problema. Evidentemente resultará diffcil manejar tres dimensiones y es del todo imposible visualizar cuatro o más. Por consiguiente, se ve la necesidad de un método totalmente numérico que aproveche la propiedad de que la o las soluciones óptimas siempre irán asociadas con vértices de la región de soluciones factibles y pueda, mediante el análisis de dichos vértices, localizar rápidamente el punto óptimo.

En particular, el Método Simplex, desarrollado por el Prof. George B. Dantzig en 1947 y que a continuación se describe, es un procedimiento notablemente eficiente para resolver el Problema General de Programación Lineal.

Como funciona el Mètodo Simplex

Los pasos que sigue el método simplex son los mismos que se siguieron en el procedimiento gráfico utilizado en la sección anterior:

Paso 1: Situarse en algún vértice de la región de soluciones factibles. Si no existe tal región, entonces el problema no tiene solución factible.

Paso 2: Determinar todas las posibles direcciones en que se podría viajar desde el vértice actual.

Paso 3: Investigar, utilizando las contribuciones de la función objetivo, si el movimiento en cada una de esas direcciones es beneficioso o perjudicial para el valor de la función objetivo.

Paso 4: a) Elegir una dirección beneficiosa y moverse en esa dirección hasta el siguiente vértice. Volver al Paso 2.

b) Si ninguna de las direcciones de movimiento puede mejorar el valor de la función objetivo, entonces el vértice actual es la solución óptima del modelo. Fin del procedimiento.

Se aprecia de inmediato que el Método Simplex es un procedimiento repetitivo, pues realiza varias veces la misma secuencia de operaciones hasta llegar al óptimo. Cada uno de estos ciclos se denomina *iteración*. El Método Simplex es un procedimiento bastante eficiente que generalmente resuelve un problema en un número pequeño de iteraciones, que pueden realizarse muy rápidamente en una computadora.

El método tiene además las siguientes ventajas:

a) Incluye un procedimiento para detectar cuándo un problema no tiene solución.

b) Incluye una prueba para detectar la existencia de soluciones no acotadas.

c) Proporciona, ya sea directamente o mediante cálculos adicionales muy sencillos, la siguiente información, que se explicará detalladamente en el siguiente apartado:

1) El valor óptimo de la función objetivo.
2) Los valores óptimos de las variables de decisión.
3) Las holguras y excesos de cada restricción.
4) Los precios sombra de los recursos.
5) Los costos reducidos de las variables de decisión.
6) Los rangos de variación de los contribuciones de la función objetivo.
7) Los rangos de variación de los Lados Derechos.

El procedimiento consiste esencialmente en la solución sucesiva de sistemas de ecuaciones lineales, elegidos de manera que cada nueva solución conlleva un valor mejorado de la función objetivo. El conocimiento de la técnica algebráica es deseable pero no es un requisito para utilizar la Programación Lineal. El área principal de interés no es la solución numérica, que puede hacerse con cualquier programa comercial, sino la formulación de modelos y la interpretación de resultados, actividades que requieren mas bien familiaridad con el sistema y una buena dosis de sentido común. De todas formas, se incluye en el Apéndice II una breve exposición de los principios algebráicos del Simplex.

El Paquete Interactivo de Programación Lineal

Como el objetivo fundamental de este texto es transmitir nociones sobre construcción de modelos e interpretación de resultados, se encomendará a la computadora la laboriosa tarea de resolver los modelos. Concretamente se utilizará P.I.P.L., un paquete desarrollado expresamente para directivos no técnicos, que desean utilizar la Programación Lineal como un apoyo para la toma de decisiones, sin preocuparse demasiado por los algoritmos de solución y otros temas de interés teórico. A partir de este punto, el análisis de los modelos se realizará en base a los reportes de resultados de P.I.P.L., aunque, por supuesto, este análisis es perfectamente aplicable a los reportes de otros paquetes.

4. El reporte de resultados y su interpretación.

No se puede hacer bastante énfasis en el hecho de que la solución óptima (valor de la función objetivo y de las variables) es apenas una pequeña parte de la información útil que puede obtenerse de un modelo lineal. En esta sección se estudian, con una interpretación geométrica, las diversas secciones del reporte de resultados del Modelo 1 generado por P.I.P.L.

REPORTE 1

EJEMPLO DE PROBLEMA DE TOMA DE DECISIONES

MODELO 1 : DULCERIA LOPEZ

OBJETIVO : MAXIMIZAR LA UTILIDAD POR VENTA DE CHOCOLATES

SOLUCION OPTIMA ENCONTRADA DESPUES DE

0 ITERACIONES DE FASE I
3 ITERACIONES DE FASE II

DEL METODO SIMPLEX REVISADO.

LOS RESULTADOS NO LISTADOS EN LAS SECCIONES 2-5 VALEN CERO.

1 - VALOR MAXIMO DE LA FUNCION OBJETIVO

Utilidad 40000.000

2 - VALORES OPTIMOS DE LAS VARIABLES DE DECISION

X (1)	Dados	25.000000
X (2)	Dados R	10.000000

3 - HOLGURA (+) Y EXCESO (-) EN LAS RESTRICCIONES

3.- + Crema	.50000000

4 - VALORES DE LAS VARIABLES DUALES

1.- ChocMesa	625.00000
2.- Jarabe	2500.0000

5 - COSTOS REDUCIDOS DE LAS VARIABLES DE DECISION

6- RANGOS DE VARIACION DE LOS COSTOS DE LA FUNCION OBJETIVO

VARIABLE	LIMITE INFERIOR	ACTUAL	LIMITE SUPERIOR
X (1) Dados	600.000	1000.00	3000.00
X (2) Dados R	500.000	1500.00	2500.00

7 - RANGOS DE VARIACION DE LOS LADOS DERECHOS

RESTRICCION	LIMITE INFERIOR	ACTUAL	LIMITE SUPERIOR
1.- ChocMesa	16.0000	24.0000	40.0000
2.- Jarabe	6.00000	10.0000	12.0000
3.- Crema	1.00000	1.50000	Ilimitado

---- o ----

Valores óptimos de la función objetivo y las variables

Estos valores son ordinariamente las primeras secciones en los listados de resultados de los paquetes comerciales de PL. No requieren mayor explicación, aunque vale la pena comentar que los tomadores de decisiones le dan relativamente poca importancia a la solución óptima *per se*, pues, en el mejor de los casos, sólo es una buena aproximación del sistema real. Son mucho más interesantes los demás resultados, indicadores de la sensibilidad de la solución óptima a cambios en el modelo, pues representan una gran ayuda para profundizar en el conocimiento del sistema bajo análisis.

Holgura y exceso en las restricciones

Cuando en el óptimo alguna de las restricciones de desigualdad ($\leq$ ó $\geq$) se cumple como igualdad estricta se dice que esa es una restricción activa. Una restricción inactiva es aquella que en el óptimo se cumple como desigualdad, esto es, que tiene una holgura o un excedente positivo.

Formalmente, para una restricción " $\leq$ " de la forma

$$\sum_{i=1}^{n} a_i x_i \leq b \; ;^{*}$$

se define la *holgura* de la restricción de la siguiente manera:

$$\text{Holgura} = b - \sum_{i=1}^{n} a_i x_i \; ;$$

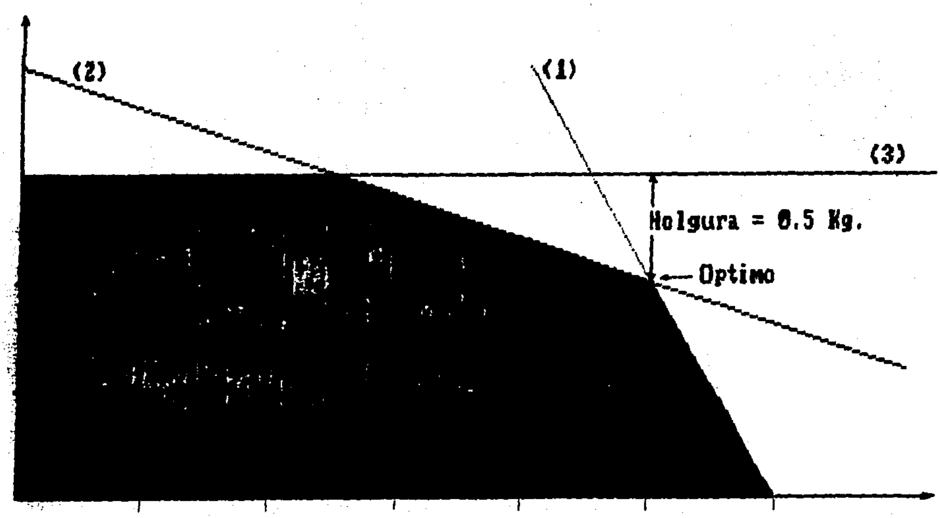
---

[*] El símbolo Σ es simplemente una manera compacta de representar una larga suma de términos. Concretamente:

$$\sum_{i=1}^{n} a_i x_i = a_1 x_1 + a_2 x_2 + \ldots + a_n x_n \; .$$

donde las x_j son los valores óptimos de las variables de decisión. Por ejemplo, volviendo al ejemplo del Sr. López, la sección 3 del listado de resultados indica que, en la solución óptima, la tercera restricción está inactiva, puesto que ha sobrado 0.5 Kg. de crema batida. Esto se confirma en la Gráfica 4, donde además puede apreciarse claramente cómo esta restricción no influye en la determinación del punto óptimo de producción, que se encuentra en el cruce de las rectas que representan la disponibilidad máxima de Chocolate de Mesa y Jarabe de Chocolate.

GRAFICA 4



Esto significa que ambos recursos son totalmente utilizados en el plan óptimo de producción. En otras palabras, ambas restricciones son activas y sus holguras respectivas valen cero.

Para una restricción " $\geq$ " de la forma

$$\sum_{j=1}^{n} a_j x_j \geq b ;$$

se define el *exceso* de la restricción como

$$\text{Exceso} = \sum_{j=1}^{n} a_j x_j - b .$$

Por ejemplo, para una restricción de calidad que exige que un cierto producto contenga un mínimo contenido de vitamina C, medida en miligramos, el exceso representaría la cantidad de vitamina que el producto tiene *por encima* del requerimiento.

Precios sombra

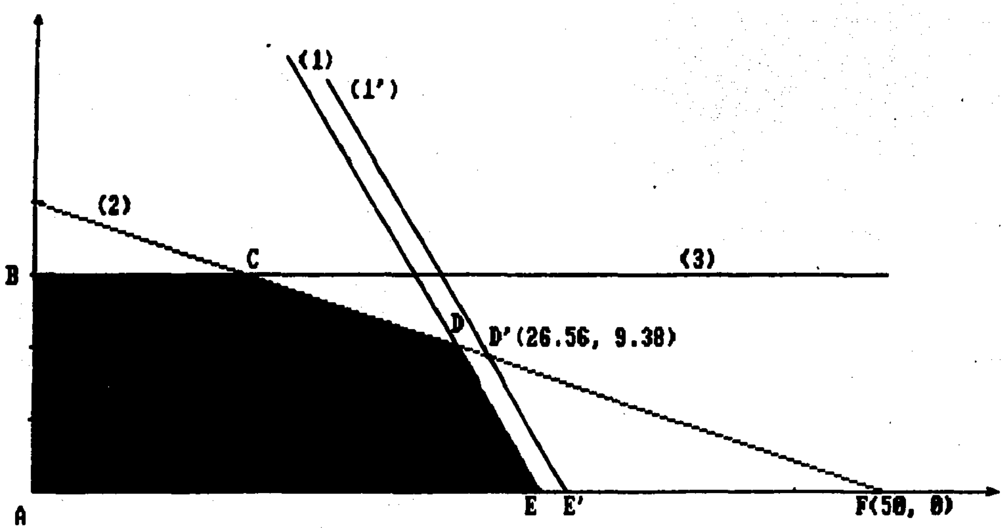
Una de las tareas más difíciles en la asignación de recursos escasos a distintas actividades productivas, es la evaluación del *beneficio neto* generado por la obtención de una unidad adicional de alguno de dichos recursos. En Economía este valor se conoce con el nombre de *Precio Sombra* del recurso. El cálculo de este valor es complicado, pues sólo puede hacerse después de averiguar cuál será el uso óptimo de esta unidad extra. Es evidente que este dato tiene un valor decisivo como criterio para la adquisición de recursos adicionales: esta conviene sólo si el precio sombra es mayor que el precio de compra. Otra aplicación se presenta cuando se está considerando la adición al sistema de una nueva actividad productiva. El costo económico real de los recursos que consumiría será la suma de los precios sombra de dichos recursos. Al comparar este con la contribución de la nueva actividad a la función objetivo, se obtiene un resultado claro de costo-beneficio, que servirá para decidir si es o no conveniente arrancar la nueva actividad.

Las técnicas de PL hacen sencilla la tarea de calcular los precios sombra, y la mayor parte de los paquetes computacionales incluyen esta información en sus listados de resultados. Algunos de ellos, como P.I.P.L., proporcionan en dichos listados una sección llamada "Valores óptimos de las variables duales". No hay que dejarse impresionar por este tecnicismo matemático. El valor óptimo de una variable dual indica *el cambio en el valor óptimo de la función objetivo, que sería ocasionado por un aumento unitario en el lado derecho de la restricción correspondiente.* En el caso de que la restricción sea la disponibilidad de un recurso, esta definición coincide con la de precio sombra del recurso. Un ejemplo puede aclarar todo esto.

Considérese la sección 4 del reporte de resultados del Modelo 1 junto con la correspondiente Gráfica 5; concretamente la restricción de Chocolate de Mesa. Lo que esta sección indica es que un kilogramo adicional de este recurso aumentaría el valor óptimo de la función objetivo de \$40,000 a \$40,625. La gráfica lo confirma; el efecto de aumentar la cantidad disponible de Chocolate se traduce en un desplazamiento de la recta (1) hacia la derecha. El punto óptimo se traslada también, siguiendo la recta CD, hasta detenerse en el nuevo óptimo D´. El nuevo plan óptimo de producción es x = 26.56, y = 9.38; esto es, un aumento de 1.56 en x y una disminución correspondiente de 0.62 en y. El cambio en el valor de la función objetivo está dado por:

$$\text{Cambio en } z = 1,000 \times 1.56 - 1,500 \times 0.62 = \$625 \;;$$

GRAFICA 5



y el nuevo valor de la función objetivo es $z = \$40,625$. En resumen, la compra de un kilogramo adicional de Chocolate de Mesa significa una utilidad *marginal* de \$625. Por tanto, si el Sr. López estuviera interesado en aumentar sus compras de Chocolate, este es el máximo *sobreprecio* que estaría dispuesto a pagar por cada kilogramo adicional.* Pagar más le ocasionaría una reducción neta en sus utilidades.

Si continúa el desplazamiento hacia la derecha de la recta DE, esto es, si se adquieren cantidades adicionales de Chocolate de Mesa, el valor de la función objetivo seguirá creciendo a razón de \$625 por cada kilogramo adicional de este ingrediente. El proceso, sin embargo, no puede continuar indefinidamente, como se ve claramente en la gráfica 5. Nótese que el óptimo se mantendrá sobre la intersección de las rectas (1´) y (2), pero se detendrá finalmente en el punto F, que corresponde a 16 Kg. adicionales de Chocolate, sumando en total 40 Kg. Si la recta (1´) se desplaza aún más hacia la derecha, ello ya no tiene ningún efecto sobre la solución óptima ni sobre la región de soluciones factibles, que de hecho quedaría definida sólamente

---

* Nótese que se desconoce el precio de venta normal del Chocolate de Mesa. Puede suponerse, sin embargo, que ya ha sido considerado al calcular las utilidades en la función objetivo. Por tanto, el precio sombra de \$625 es el máximo *aumento* en el precio de venta del Chocolate que podría aceptar el Sr. López sin incurrir en pérdidas.

por las rectas (2), (3) y las restricciones de no negatividad x, $y \geq 0$. El problema estriba en que se ha agotado totalmente el jarabe de chocolate y, faltando este, no se podría producir nada con el 27° Kg. adicional de Chocolate de Mesa. En otras palabras, mientras que el 26° Kg. adicional podría transformarse en producto terminado y venderse con una utilidad de $625, el 27° no podría utilizarse, y por tanto el Sr. López no estaría dispuesto a comprarlo; por lo menos no sin comprar más Jarabe de Chocolate. Por eso el precio sombra del Chocolate de Mesa se vuelve $0 al rebasar los 40 Kg.

Por lo expuesto anteriormente, hay que aclarar acerca de los precios sombra que son válidos solamente en un cierto rango de variación alrededor de la disponibilidad actual del recurso considerado, y cambian (no necesariamente a cero) cuando la variación sale de este rango. Los paquetes de Programación Lineal dedican una sección de sus reportes de resultados a esta información; en P.I.P.L., es la sección 7. Obsérvese, en el Reporte 1, que el límite superior de la restricción de Chocolate de Mesa es, efectivamente, 40.

Por otro lado, resulta claro, intuitivamente, que el precio sombra de una restricción no activa es cero, pues si en el óptimo el recurso no fué totalmente consumido, conseguir una cantidad adicional no tendrá ningún efecto sobre la función objetivo. Por ejemplo, la restricción 3, correspondiente a la crema batida, no apareció en la sección 4, pues su precio sombra es cero, y la sección 7 no señala ningún límite superior. Se concluye que el obtener más crema no cambiará la solución actual.

Costos Reducidos de las Variables de Decisión

Las restricciones de no negatividad tienen también precios sombra, pero estos reciben el nombre más adecuado de *Costos Reducidos*. Si en la solución óptima alguna de las variables de decisión está a nivel cero, su costo reducido representa el perjuicio (la reducción en el valor de la función objetivo) en que se incurrirá si se decide dar a esa variable un valor mayor de cero a pesar de la recomendación del modelo --por ejemplo, en la vida real muchas veces es imposible detener totalmente una actividad--. Matemáticamente, esto equivale a cambiar la restricción de no negatividad correspondiente, $x \geq 0$ a $x \geq 1$. Otra manera de verlo sería decir que los costos reducidos son la contribución marginal (adversa), a la función objetivo, de las variables menos "rentables" del modelo. Por tanto, estos costos indican también cuánto debe aumentar la contribución de dichas variables para volverse rentables.

En el problema del Sr. López, los costos reducidos de ambas variables x y y (producción de "Dados" y "Dados Rellenos"), valen cero, ya que en el óptimo ninguna de las actividades se encuentra a nivel cero. De ahí que, en el listado de resultados de P.I.P.L., la sección 5 aparece vacía.

Sensibilidad de la solución a cambios en la función objetivo

Frecuentemente, los ingresos, costos, impuestos y demás datos que entran como coeficientes en la función objetivo son aproximaciones, no constantes conocidas. Por tanto, es esencial para el gerente o el analista saber cuál será el efecto sobre el modelo de un cambio o corrección en una de estos coeficientes.

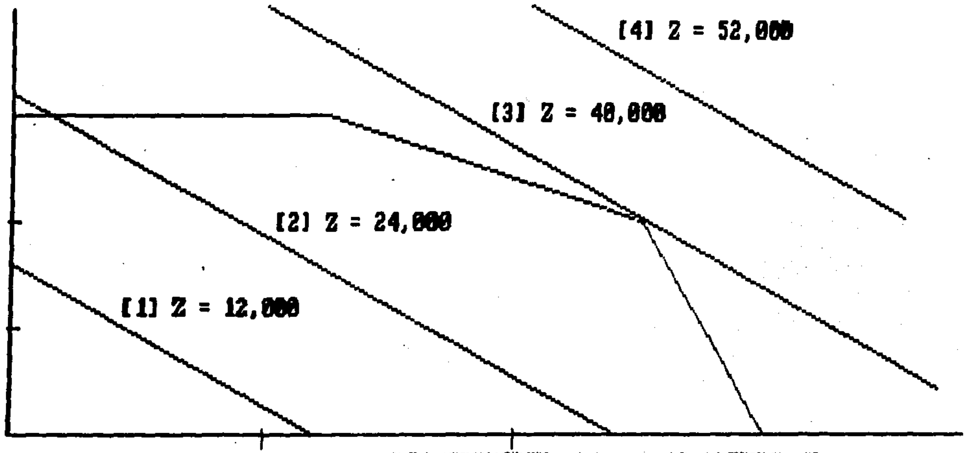
El cambio en el valor óptimo de la función objetivo, causado por una variación en la contribución de la variable *x*, está dado por la expresión

Cambio en *z* = Cambio en contribución x Valor actual de *x*.

Sin embargo, se intuye que un cambio importante en el precio de *x* puede hacer que la solución actual deje de ser óptima, y es importante saber cuál sería la máxima variación permisible antes de que esto ocurra. Esta información aparece en la sección 6 del listado de resultados y una interpretación se da en la gráfica 6.

Se pueden representar en la gráfica planes de producción diferentes, pero con un valor común de función objetivo, mediante rectas paralelas. Por ejemplo, la recta [1] está formada por todos los planes factibles de producción de chocolates que generan una utilidad constante de $12,000. La [2] representa los planes que generan una utilidad de $24,000. En la medida que estas líneas están más alejadas del origen (0, 0), conllevan niveles de utilidad mayores. Por supuesto, existe un límite: la

GRAFICA 6

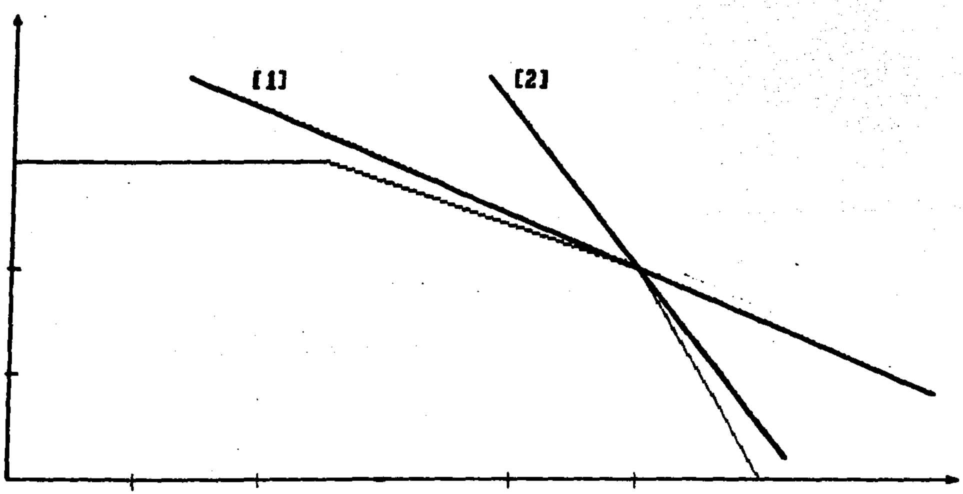


región de soluciones factibles. Por ejemplo, la recta [4] tiene una utilidad de $52,000, pero ninguno de los correspondientes planes de producción --representados por los puntos que forman la recta-- puede llevarse a cabo con los recursos disponibles. Por lo tanto, para encontrar el óptimo, lo que se hace es trazar rectas paralelas cada vez más alejadas del origen, buscando el último punto de la región factible que estas rectas pueden tocar --intuitivamente es claro que ese punto será un vértice-- antes de pasar a la región de no factibilidad, y ese será el nivel máximo de utilidad, representado aquí por la recta [3].

Ahora nótese en la gráfica 7 que, si se toma el vértice óptimo D como punto de apoyo, la recta que representa el valor máximo de la función objetivo puede cambiar considerablemente su pendiente, en ambos sentidos, sin tocar ningún otro punto de la región de soluciones factibles; esto es, *sin que cambie el vértice óptimo.* Algebráicamente, este cambio de pendiente ocurre cuando los coeficientes de la función objetivo cambian uno respecto del otro. Las rectas [1] y [2] en la gráfica 7 corresponden a las siguientes funciones objetivo:

$$[1] : \quad z = 1{,}000\, x + 2{,}000\, y$$
$$[2] : \quad z = 2{,}250\, x + 1{,}500\, y$$

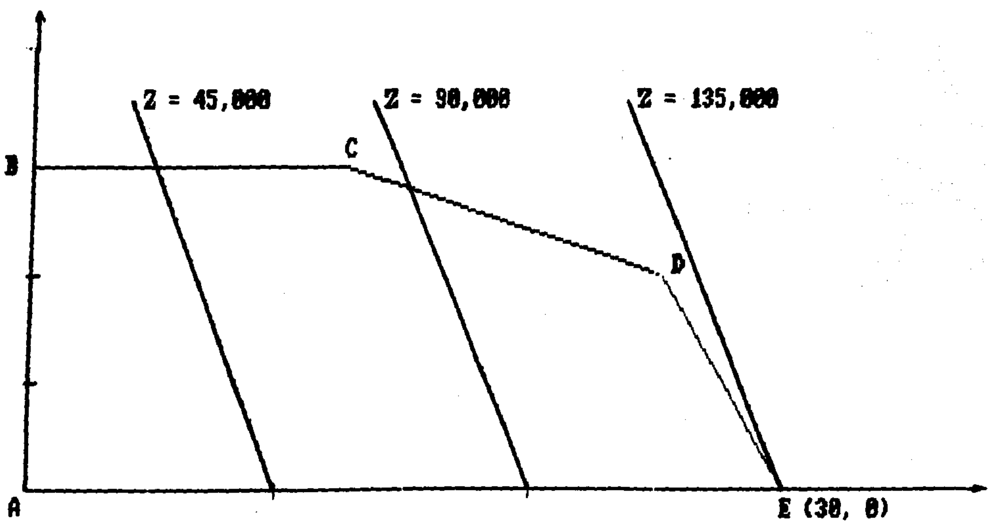
GRAFICA 7



Véase como, aunque se trata de funciones muy distintas a la inicial, el vértice óptimo sigue siendo el mismo (aunque cambie el valor de z), pues es el último punto de la región de soluciones que las funciones objetivo tocan al alejarse del origen buscando el máximo. Se mantienen fijos los valores óptimos de las variables de decisión, pero cambia el valor de z.

Cuando el cambio en los coeficientes es aún mayor ocurre lo que se muestra en la Gráfica 8, donde aparece la función objetivo $z = 4,500\,x + 1,500\,y$, para diferentes niveles de utilidad.

GRAFICA 8



Esta recta tiene una pendiente tal, que el óptimo ha pasado del vértice D al E.

Los listados de resultados de los paquetes de PL incluyen normalmente una sección que señala, para cada variable, cuánto puede cambiar su contribución en ambos sentidos sin que cambie la solución óptima. En P.I.P.L. es la seccion 6, y puede verificarse en el Reporte 1 que la función objetivo de la Gráfica 8 tiene una contribución para x que ha salido del rango de variación permitido. Es por eso que la solución óptima se ha cambiado al vértice E.

Esto concluye la exposición de la teoría básica de Programación Lineal; se tienen ya los instrumentos necesarios para abordar, en la segunda parte del manual, el extenso campo de las aplicaciones prácticas.

SEGUNDA PARTE : APLICACIONES DE PROGRAMACION LINEAL

Esta sección presenta ejemplos de tres clases de problemas que han sido exitosamente resueltos mediante Programación Lineal: El Problema de Selección de Cartera de Inversión, el Problema de Asignación de Personal y el Problema de la Dieta. Con estos ejemplos se pretende ejercitar el material de la primera parte y mostrar la gran diversidad de situaciones en que la PL puede actuar eficazmente. Dirigidos, como todo el texto, al tomador de decisiones, pretenden además sugerir nuevas posibilidades de aplicación.

Capítulo III

Un Problema de Selección de Cartera de Inversión.

Caso 2: González Hnos., S. A.

González Hermanos, S. A. es un distribuidor de materiales de construcción de Guadalajara, Jalisco. Tiene varios clientes importantes que con frecuencia colocan pedidos de diez o veinte millones de pesos y pagan de inmediato para aprovechar el descuento del 10% que la compañía otorga por pagos al contado.

Jose Luis García, el contralor de la empresa, lleva un par de semanas estudiando el problema de cómo aprovechar los excesivos saldos en cuenta de cheques que este tipo de ventas ocasionan constantemente. Con una inflación anual del 90%, García está decidido a mantener un saldo promedio mensual de 1 millón en la chequera, en vez de los 12 a 18 millones que se han tenido hasta ahora.

El plan que ha formulado después de varias consultas con el dueño de la compañía, el Sr. Ramón González, consiste en formar una pequeña cartera con cinco opciones distintas de inversión, cuyas características se muestran en el siguiente cuadro:

	Rendimiento mensual esperado (%)	Indice de Riesgo
Plazo Fijo	6.125	1
CETES	6.0	3
Papel Comercial	6.2	3
Sociedad de Inversión	6.75	4
Acciones comunes	7.875	6

a) Las inversiones a plazo fijo son por 30 días.

b) El papel comercial puede venderse cinco días después de su adquisición.

c) Los CETES, las acciones comunes y las acciones de la sociedad de inversión tienen liquidez inmediata.

El índice de riesgo es una convención entre García y el Sr. González, para comparar los riesgos de distintos planes de inversión. Un índice de riesgo de 1, a juicio del Sr. González, indica una inversión prácticamente segura y un índice de 6 señala el máximo riesgo que está dispuesto a correr.

El Sr. González ha establecido tres criterios para la administración de la cartera de inversión:

1) Que en cualquier momento se pueda sacar $1 millón "para lo que se ofrezca."

2) Que en un plazo no mayor de una semana se pueda disponer de la mitad del dinero invertido.

3) Que el índice de riesgo promedio de la cartera no pase de 3 en la escala convenida. Este índice se calcula sumando las cantidades invertidas en cada opción, multiplicadas por su índice de riesgo correspondiente y dividiendo el total entre el monto de la cartera.

El contralor ha decidido hacer una inversión inicial de diez millones de pesos durante los primeros 30 días del proyecto.

— o —

Para plantear este problema de decisión como un modelo lineal se siguen los pasos detallados en el capítulo anterior. En primer lugar se deben definir las variables de decisión. Es obvio que la pregunta que se hace el Sr. García es cuánto dinero invertir en cada opción. Por consiguiente, se pueden definir las variables de la siguiente manera:

Num. Var.	Clave	Definición
X_1	PlazoFij	Cantidad a invertir a plazo fijo.
X_2	CETES	Cantidad a invertir en CETES.
X_3	PapelCom	Cantidad a invertir en papel comercial.
X_4	Sociedad	Inversión en acciones de la sociedad.
X_5	Acciones	Inversión en acciones comunes.

El objetivo del Sr. García es sencillamente obtener el mayor rendimiento posible de la cartera, por lo que la función objetivo puede escribirse de inmediato:

$$\text{Max} \quad z = .06125\, X_1 + .06\, X_2 + .062\, X_3 + .0675\, X_4 + .07875\, X_5 .$$

Por último, se plantean las restricciones del modelo: las tres condiciones impuestas por el Sr. González, y el monto inicial de la cartera elegido por el Sr. García. Esta última restricción es la más sencilla y se escribe de inmediato:

$$X_1 + X_2 + X_3 + X_4 + X_5 \leq 10'000,000 .$$

Ahora bien, la primera condición dice verbalmente que debe haber por lo menos $1 millón en inversiones liquidables de inmediato. Las opciones que poseen esta característica son X_2, X_4 X_5, por lo que la restricción toma la forma:

$$X_2 + X_4 + X_5 \geq 1'000,000 .$$

La segunda condición exige que la mitad del monto de la cartera pueda retirarse en menos de una semana. Obsérvese que todas las opciones de inversión, salvo los depósitos a plazo fijo, cumplen con esta condición. Por tanto, se puede escribir:

Capital retirable en 1 semana $\geq$ 1/2 (Monto de la cartera),

$$X_2 + X_3 + X_4 + X_5 \geq 1/2\ (X_1 + X_2 + X_3 + X_4 + X_5) .$$

Multiplicando ambos lados de la expresión por 2 para eliminar (por comodidad) la fracción, se obtiene:

$$2X_2 + 2X_3 + 2X_4 + 2X_5 \geq X_1 + X_2 + X_3 + X_4 + X_5 .$$

Despejando esta expresión para dejar todas las variables del lado izquierdo y una constante del lado derecho, como lo exige el formato habitual, se obtiene la tercera restricción del modelo:

$$- X_1 + X_2 + X_3 + X_4 + X_5 \geq 0 .$$

Nótese que también se habría podido expresar esta condición con la restricción $X_2 + X_3 + X_4 + X_5 \geq 5'000,000$. Sin embargo como es probable que cada mes cambie la cantidad invertida, la primera restricción es preferible, pues no depende del monto de la cartera y hace más fácil el actualizar mensualmente el modelo.

Por último se pide que el índice de riesgo promedio de la cartera no exceda a 3. Según la definición dada esto significa:

$$\frac{X_1 + 3X_2 + 3X_3 + 4X_4 + 6X_5}{X_1 + X_2 + X_3 + X_4 + X_5} \leq 3 .$$

Multiplicar ambos lados por $(X_1 + X_2 + X_3 + X_4 + X_5)$ y mover todas las variables al lado izquierdo produce la siguiente restricción *lineal*:

$$-2\,X_1 + X_4 + 3\,X_5 \leq 0 .$$

En resumen, se obtiene el modelo lineal y reporte de resultados que se listan a continuación:

Modelo 2 : González Hnos, S. A.

Determinar las cantidades a invertir en cada opción de la cartera para

$$\text{Max } z = .06125X_1 + .06X_2 + .062X_3 + .0675X_4 + .07875X_5$$

sujeta a

X_1	+	X_2	+	X_3	+	X_4	+	X_5	$\leq$	10000000	Capital
		X_2			+	X_4	+	X_5	$\geq$	1000000	Inmediato
$-X_1$	+	X_2	+	X_3	+	X_4	+	X_5	$\geq$	0	1 Semana
$-2X_1$					+	X_4	+	$3X_5$	$\leq$	0	Riesgo

$$X_1,\ X_2,\ X_3,\ X_4,\ X_5 \geq 0$$

REPORTE 2

PROBLEMA DE SELECCION DE CARTERA DE INVERSION

CASO 2 : GONZALEZ HNOS., S. A.

OBJETIVO : MAXIMIZAR RENDIMIENTO MENSUAL DE LA CARTERA

SOLUCION OPTIMA ENCONTRADA DESPUES DE

2 ITERACIONES DE FASE I
3 ITERACIONES DE FASE II

DEL METODO SIMPLEX REVISADO.

LOS RESULTADOS NO LISTADOS EN LAS SECCIONES 2-5 VALEN CERO.

1 - VALOR MAXIMO DE LA FUNCION OBJETIVO

RenMensl	672083.33

2 - VALORES OPTIMOS DE LAS VARIABLES DE DECISION

X (1)	PlazoFij	5000000.0
X (3)	PapelCom	1666666.7
X (5)	Acciones	3333333.3

3 - HOLGURA (+) Y EXCESO (-) EN LAS RESTRICCIONES

2.- - Inmediat	2333333.3

4 - VALORES DE LAS VARIABLES DUALES

1.- Capital	6.72083333E-02
3.- 1 Semana	-5.20833333E-03
4.- Riesgo	5.58333333E-03

5 - COSTOS REDUCIDOS DE LAS VARIABLES DE DECISION

2.- CETES	-2.00000000E-03
4.- Sociedad	-8.33333333E-05

6- RANGOS DE VARIACION DE LOS COSTOS DE LA FUNCION OBJETIVO

VARIABLE		LIMITE INFERIOR	ACTUAL	LIMITE SUPERIOR
X (1)	PlazoFij	5.083333E-02	6.125000E-02	Ilimitado
X (2)	CETES	Ilimitado	6.000000E-02	6.200000E-02
X (3)	PapelCom	6.187500E-02	6.200000E-02	6.825000E-02
X (4)	Sociedad	Ilimitado	6.750000E-02	6.758333E-02
X (5)	Acciones	7.850000E-02	7.875000E-02	Ilimitado

7 -. RANGOS DE VARIACION DE LOS LADOS DERECHOS

RESTRICCION		LIMITE INFERIOR	ACTUAL	LIMITE SUPERIOR
1.-	Capital	3.000000E+06	1.000000E+07	Ilimitado
2.-	Inmediat	Ilimitado	1.000000E+06	3.333333E+06
3.-	1 Semana	-2.000000E+06	.000000	7.000000E+06
4.-	Riesgo	-7.000000E+06	.000000	5.000000E+06

Además de los valores óptimos de inversión y el rendimiento máximo mensual, este reporte contiene información de gran utilidad para García. La sección 3 le informa que hay mas que suficiente capital liquidable de inmediato, pero, en cambio, el requerimiento de 1 semana se cumple estrictamente (la restricción 3 no aparece en la sección de Holguras y Excesos). Por otro lado, el precio sombra de esta última restricción, es - 0.005208 . Esto indica que relajar esta condición, permitiendo que sea una cantidad menor la que esté invertida en opciones liquidables antes de una semana, y aumentando de esta manera la inversión a plazo fijo, produciría un rendimiento marginal de 0.005208. Esto es una sorpresa, pues el plazo fijo es el que tiene menor interés entre las opciones de inversión recomendadas por el paquete. Lo que ocurre es que no se puede invertir más dinero en las otras dos, Papel Comercial y Acciones, porque la restricción de Riesgo ya está en su límite, como se sabe por no estar listada en la sección de holguras y porque su precio sombra es positivo.

La sección 5, referente a las variables que están a nivel de cero, indica que un aumento de 0.002 en el rendimiento de los CETES haría que esta opción entrara en la solución óptima, mientras que bastaría un minúsculo incremento de 0.0000833 para volver rentables las acciones de la Sociedad de Inversión. Estas observaciones se confirman en la sección 6. Obsérvese que las contribuciones de las tres últimas inversiónes están todas muy cercanas a alguno de sus dos límites máximos de variación. Por ejemplo, una disminución de .00025 en el rendimiento de las Acciones comunes bastaría para que cambiara la solución óptima, seguramente reemplazando esta inversión con alguna de las que aparecen en la sección 5. En resumen, la solución óptima actual es muy sensible a cambios en los rendimientos.

Esto hace del modelo de Programación Lineal una estupenda herramienta para decidir rápidamente qué opciones de inversión deben seguirse en un mercado con tasas de interés poco estables.

Capítulo IV

Un Problema de Asignación de Personal

Caso 3: Grande, S. A.

Grande, S. A. es una cadena de tiendas de autoservicio que cuenta con numerosas sucursales en la Ciudad de México. De ellas, la tienda Norte, la más importante de la compañía, se cuenta entre los almacenes de autoservicio más grandes del país.

Recientemente el Sr. Carlos Salas, Vicepresidente de Personal de la compañía, hizo una visita a la sucursal Norte y notó que había en ese momento más cajas registradoras en servicio de las que realmente se necesitaban. Grande tiene una reputación arduamente ganada de hacer esperar lo menos posible al cliente en las cajas registradoras (de hecho esta sucursal cuenta con 25 cajas). Sin embargo, el Sr. Salas quiso averiguar que criterios se utilizaban para la asignación de turnos al personal de cajas.

En resumen, la información que obtuvo del departamento de personal de la tienda es la siguiente:

1) La tienda emplea dos tipos de cajeras, de medio tiempo y de tiempo completo.

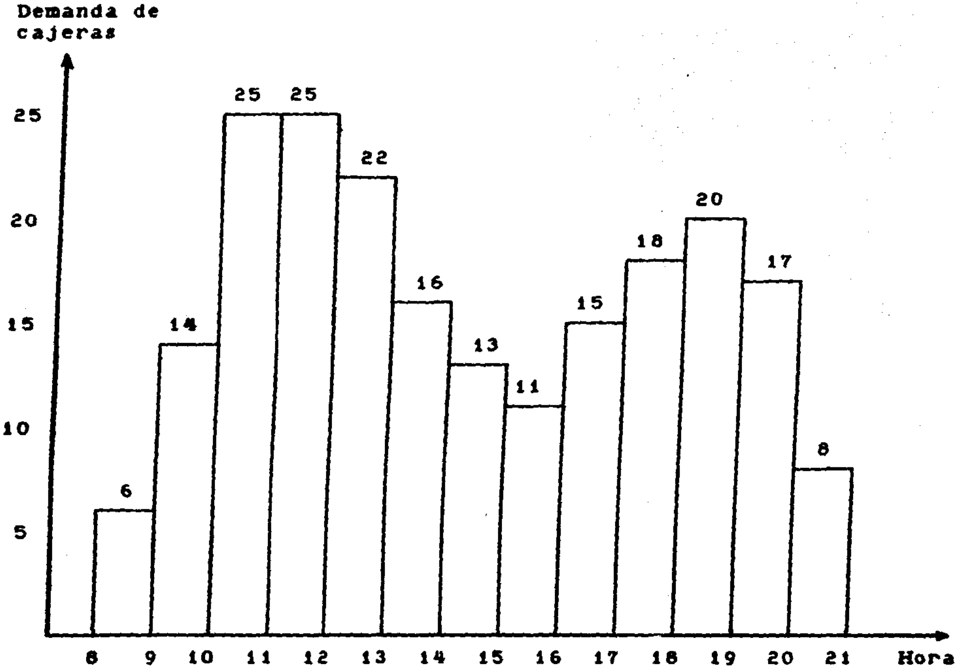
2) Las cajeras de tiempo completo trabajan un turno de cuatro horas, después tienen una hora para comer y terminan con otro turno de cuatro horas. Las cajas estan abiertas de 8 A.M. a aproximadamente 8:35 P.M. (las puertas se cierran a las 8 P.M., pero siempre quedan clientes por atender). Por tanto una cajera de tiempo completo debe entrar a trabajar a más tardar a las 12 P.M. Este personal le ocasiona a la tienda un costo de $525 por hora (incluyendo la hora de comida), que incluye salario, otras prestaciones e impuestos.

3) Las cajeras de medio tiempo trabajan un solo turno de cuatro horas y tienen un costo de $635 por hora.

4) Después de varios meses de experiencia se ha podido elaborar una serie de pronósticos de demanda de cajeras para diferentes días del mes y de la semana. La Gráfica 9 muestra uno de ellos. Estos pronósticos han demostrado ser acertados el 90% del tiempo y en ellos se basa el departamento de personal para la asignación de turnos de cajeras.

El Sr. Salas sospecha que se pueden reducir los costos de personal de cajas y pide al gerente de personal de la tienda que determine para cada pronóstico la estrategía óptima de asignación de turnos para minimizar el costo diario de la tienda.

GRAFICA 9



—— o ——

La decisión que debe tomar el gerente de personal comprende dos aspectos: cuántas cajeras contratar y qué turnos se les deben asignar. Existen dos tipos de cajeras y trece horas de entrada distintas a lo largo del día, de las 8 A.M. a las 17 P.M. Para que el modelo lineal elija la mejor asignación posible de turnos debe "conocer" todas las opciones disponibles, esto es, todas las posibles combinaciones de dos tipos de cajeras y trece horas de llegada. Por consiguiente, se pueden definir las variables de decisión de la siguiente manera:

Num.	Clave	Definición
X_1	Medio 8	Num. cajeras medio tiempo que entran a las 8
X_2	Compl 8	Num. cajeras tiempo completo, entran a las 8
X_3	Medio 9	Num. cajeras medio tiempo, entran a las 9
X_4	Compl 9	Num. cajeras tiempo completo, entran a las 9
X_5	Medio 10	Num. cajeras medio tiempo, entran a las 10
X_6	Compl 10	Num. cajeras tiempo completo, entran a las 10
X_7	Medio 11	Num. cajeras medio tiempo, entran a las 11
X_8	Compl 11	Num. cajeras tiempo completo, entran a las 11
X_9	Medio 12	Num. cajeras medio tiempo, entran a las 12
X_{10}	Compl 12	Num. cajeras tiempo completo, entran a las 12
X_{11}	Medio 13	Num. cajeras medio tiempo, entran a las 13
X_{12}	Medio 14	Num. cajeras medio tiempo, entran a las 14
X_{13}	Medio 15	Num. cajeras medio tiempo, entran a las 15
X_{14}	Medio 16	Num. cajeras medio tiempo, entran a las 16
X_{15}	Medio 17	Num. cajeras medio tiempo, entran a las 17

Nótese que una vez que se han definido las variables los demás pasos en la construcción del modelo son casi automáticos. Para escribir la función objetivo, únicamente se necesita calcular el costo diario de una cajera de tiempo completo y una de medio tiempo. Esto es:

Personal Medio Tiempo : 4 hrs. x \$635 por hr. = \$2540.

Personal Tiempo Completo : 9 hrs. x \$525 por hr. = \$4725.

Por consiguiente, los costos de la función objetivo serán \$4725 para las variables X_2, X_4, X_6, X_8 y X_{10}, que corresponden a personal de tiempo completo, y \$2540 para las demás variables, que representan personal de medio tiempo.

La condición que debe cumplir cualquier propuesta de cajeras y turnos es que se cubra la demanda de cada hora en las cajas registradoras. Por lo tanto, el modelo tendrá en total trece restricciones, las cuales pueden escribirse, en términos de las variables definidas, de la siguiente manera:

Primera restricción, demanda de las 8 a las 9 A.M. :

$$X_1 + X_2 \geq 6 .$$

Segunda restricción, demanda de las 9 a las 10 A.M. :

$$X_1 + X_2 + X_3 + X_4 \geq 14 .$$

Y de igual manera las demás. Hay que tener en cuenta que las cajeras de tiempo completo tienen una hora para comer y durante ese tiempo no podrán ayudar a cubrir la demanda en las cajas registradoras. Es la razón de que la variable X_2, por ejemplo, tenga coeficientes 1.0 en las restricciones 1 a 9, exceptuando la restricción 5 (turno de 12 a 13) donde su coeficiente es cero. El modelo completo puede escribirse tabularmente de la siguiente manera:

Modelo 3: Supermercado Grande, S.A.

$$\begin{aligned} \text{Min } z = {} & 2540\ X_1 + 4725\ X_2 + 2540\ X_3 + 4725\ X_4 + 2540\ X_5 \\ & + 4725\ X_6 + 2540\ X_7 + 4725\ X_8 + 2540\ X_9 + 4725\ X_{10} \\ & + 2540\ X_{11} + 2540\ X_{12} + 2540\ X_{13} + 2540\ X_{14} + 2540\ X_{15} \end{aligned}$$

Restricciones

V a r i a b l e s																	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		L. D.	Clave de Restric.
1	1														≥	6	8 — 9
1	1	1	1												≥	14	9 — 10
1	1	1	1	1	1										≥	25	10 — 11
1	1	1	1	1	1	1	1								≥	25	11 — 12
		1	1	1	1	1	1	1	1						≥	22	12 — 13
	1			1	1	1	1	1	1	1					≥	16	13 — 14
	1		1			1	1	1	1	1	1				≥	13	14 — 15
	1		1		1			1	1	1	1	1			≥	11	15 — 16
	1		1		1		1			1	1	1	1		≥	15	16 — 17
			1		1		1		1		1	1	1	1	≥	18	17 — 18
					1		1		1			1	1	1	≥	20	18 — 19
							1		1				1	1	≥	17	19 — 20
									1					1	≥	8	20—20:35

Todas las variables no negativas.

— o —

El reporte de resultados del Modelo 3 generado por P.I.P.L. sigue a continuación:

REPORTE 3

PROBLEMA DE ASIGNACION DE TURNOS DE CAJERAS

CASO 3: SUPERMERCADO GRANDE, S. A.

OBJETIVO : MINIMIZAR COSTO DIARIO DEL PERSONAL DE CAJAS

SOLUCION OPTIMA ENCONTRADA DESPUES DE

15 ITERACIONES DE FASE I
1 ITERACIONES DE FASE II

DEL METODO SIMPLEX REVISADO.

LOS RESULTADOS NO LISTADOS EN LAS SECCIONES 2-5 VALEN CERO.

1 - VALOR MINIMO DE LA FUNCION OBJETIVO

COSTOS 141640.00

2 - VALORES OPTIMOS DE LAS VARIABLES DE DECISION

X (2)	COMPL 8	8.0000000
X (3)	MEDIO 9	6.0000000
X (5)	MEDIO 10	8.0000000
X (6)	COMPL 10	3.0000000
X (8)	COMPL 11	4.0000000
X (10)	COMPL 12	1.0000000
X (15)	MEDIO 17	12.000000

3 - HOLGURA (+) Y EXCESO (-) EN LAS RESTRICCIONES

1.-	- 8 - 9	2.0000000
4.-	- 11 - 12	4.0000000
6.-	- 13 - 14	8.0000000
8.-	- 15 - 16	1.0000000
10.-	- 17 - 18	2.0000000
13.-	- 20-20:35	5.0000000

4 - VALORES DE LAS VARIABLES DUALES

3.-	10 - 11	2540.0000
7.-	14 - 15	2185.0000
11.-	18 - 19	2185.0000
12.-	19 - 20	355.00000

5 - COSTOS REDUCIDOS DE LAS VARIABLES DE DECISION

7.-	MEDIO 11	355.00000
9.-	MEDIO 12	355.00000
11.-	MEDIO 13	355.00000
12.-	MEDIO 14	355.00000
13.-	MEDIO 15	355.00000

6 - RANGOS DE VARIACION DE LOS COSTOS DE LA FUNCION OBJETIVO

VARIABLE		LIMITE INFERIOR	ACTUAL	LIMITE SUPERIOR
X (1)	MEDIO 8	2540.00	2540.00	Ilimitado
X (2)	COMPL 8	355.000	4725.00	4725.00
X (3)	MEDIO 9	2540.00	2540.00	2540.00
X (4)	COMPL 9	4725.00	4725.00	Ilimitado
X (5)	MEDIO 10	2185.00	2540.00	2540.00
X (6)	COMPL 10	2540.00	4725.00	5080.00
X (7)	MEDIO 11	2185.00	2540.00	Ilimitado
X (8)	COMPL 11	4725.00	4725.00	4725.00
X (9)	MEDIO 12	2185.00	2540.00	Ilimitado
X (10)	COMPL 12	4725.00	4725.00	4725.00
X (11)	MEDIO 13	2185.00	2540.00	Ilimitado
X (12)	MEDIO 14	2185.00	2540.00	Ilimitado
X (13)	MEDIO 15	2185.00	2540.00	Ilimitado
X (14)	MEDIO 16	2540.00	2540.00	Ilimitado
X (15)	MEDIO 17	2540.00	2540.00	2540.00

7 - RANGOS DE VARIACION DE LOS LADOS DERECHOS

RESTRICCION		LIMITE INFERIOR	ACTUAL	LIMITE SUPERIOR
1.-	8 - 9	Ilimitado	6.00000	8.00000
2.-	9 - 10	8.00000	14.0000	22.0000
3.-	10 - 11	23.0000	25.0000	33.0000
4.-	11 - 12	Ilimitado	25.0000	29.0000
5.-	12 - 13	14.0000	22.0000	24.0000
6.-	13 - 14	Ilimitado	16.0000	24.0000
7.-	14 - 15	12.3333	13.0000	21.0000
8.-	15 - 16	Ilimitado	11.0000	12.0000
9.-	16 - 17	11.0000	15.0000	16.0000
10.-	17 - 18	Ilimitado	18.0000	20.0000
11.-	18 - 19	19.5000	20.0000	24.0000
12.-	19 - 20	13.0000	17.0000	17.5000
13.-	20-20:35	Ilimitado	8.00000	13.0000

De las secciones del listado de resultados probablemente la que más interesaría al Sr. Salas, siendo este el punto que primeramente atrajo su atención, es la de holguras y excesos en las restricciones, que indicará los excedentes de personal en los diversos turnos. Sin embargo, antes de mostrarle los resultados sería conveniente vaciar la información en un reporte menos técnico y más esquemático. Una presentación en forma tabular como la mostrada en la página siguiente permite tener una visión de conjunto de los requerimientos del problema y la solución propuesta por el modelo.

El reporte tiene varios puntos interesantes. En primer lugar, a pesar de que se ha minimizado el costo, la sección 3 muestra que aún hay 6 turnos en el día en que habrá más personal en servicio del estrictamente necesario. De hecho en el turno de 11 a 12 no hay suficientes cajas registradoras para todo el personal. Nótese también como hay 8 cajeras ociosas de las 13 a las 14 horas. Sin embargo, esta es la solución de menor costo con la estructura actual de turnos y demandas por turno. Un modelo lineal parecido podría emplearse para intentar eliminar desperdicios como el de 13 a 14, ensayando, por ejemplo, la posibilidad de turnos de cinco y tres horas para algunas cajeras, además de los actuales de cuatro y cuatro.

Otro dato que seguramente sería de gran interés para al Sr. Salas es el alto costo que tiene el servicio a los clientes a ciertas horas. En concreto, la sección 4 informa que el precio

TABLA 4-1 : ASIGNACION OPTIMA DE TURNOS DE CAJERAS

X_2 COM 8	X_3 MED 9	X_5 MED10	X_6 COM10	X_8 COM11	X_{10} COM12	X_{15} COM13	Deman	Exced	Turno
8							6	2	8 - 9
8	6						14		9 -10
8	6	8	3				25		10-11
8	6	8	3	4			25	4	11-12
	6	8	3	4	1		22		12-13
8		8	3	4	1		16	8	13-14
8				4	1		13		14-15
8			3		1		11	1	15-16
8			3	4			15		16-17
			3	4	1	12	18	2	17-18
			3	4	1	12	20		18-19
				4	1	12	17		19-20
					1	12	8	5	20 -

Personal de tiempo completo requerido: 16

Personal de medio tiempo requerido: 26

Costo Mínimo Diario: 16 × 4725 + 26 × 2540 = $141,640.

sombra del turno de 10 a 11 A.M. es $2540, y la sección 7 indica que esto es válido para lados derechos desde 23 hasta 33. Hablando claramente, esto significa que el costo total de atender las cajas 23, 24 y 25, *solamente de las 10 a las 11 A.M.*, es de

2540 x 3 = $7620. En otras palabras, si el requerimiento de cajeras de este turno fuera solamente de 22, la tienda podría ahorrarse el sueldo de tres cajeras de medio tiempo. En los turnos de 14 a 15 y de 18 a 19 horas ocurre algo parecido. Disminuyendo unitariamente el requerimiento de cada uno, se ahorrarían 2 x 2185 = $4370. El modelo lineal, mediante los precios sombra, permite detectar la dramática diferencia que hay entre el costo contable ($635) de contratar una cajera de medio tiempo para atender la caja 25 de las 10 a las 11 y el costo económico real ($2540). Es evidente que información de esta naturaleza, que la contabilidad no puede proporcionar, tiene un gran valor para el Sr. Salas que podrá apoyarse en ella para tomar las decisiones que el caso requiere. Es importante enfatizar el papel exclusivamente analítico e informativo que juegan los modelos en la toma de decisiones. Tratándose de un juicio de valor, corresponde únicamente al Sr. Salas determinar si vale la pena asignar una cajera menos en el turno de 10 a 11, ahorrando una contratación de medio tiempo, a cambio de hacer esperar un poco más a los clientes a esa hora. En otras palabras, la decisión la toma el Vicepresidente de Personal, no el modelo.

Capítulo V

Un Problema de Mezcla Optima de Productos

Caso 4: Centro Agropecuario Experimental *El Peñón*

La Ex-Hacienda de Santa Clara de Montefalco esta enclavada en el Valle de Amilpas, a unos veinte minutos de Cuautla, Estado de Morelos. Antiguamente fue un floreciente ingenio azucarero en el que trabajaban centenares de personas. Después, durante la Revolución, el lugar fue quemado y abandonado.

Hoy en día, reconstruida, Montefalco es, entre otras cosas, sede del Centro Agropecuario Experimental *El Peñón*, que desde el año de 1959 ha impartido cursos para mejorar la instrucción de los campesinos, de manera que obtengan el máximo rendimiento de los cultivos: técnicas de siembras, ciclos agrícolas, sistemas de riego, ganadería, avicultura, etc.

El Lic. Manuel Téllez es el veterinario encargado de atender las cuarenta cabezas de ganado bovino de la escuela. Uno de los problemas que enfrenta es el de preparar la ración diaria de los animales. Se trata de satisfacer sus necesidades nutricionales básicas utilizando los productos alimenticios disponibles en el mercado. En las tablas 5-1 y 5-2 se resumen los datos relevantes.

TABLA 5-1 : PRODUCTOS ALIMENTICIOS PARA GANADO BOVINO

	Contenido alimenticio (% por peso)			
Producto	Proteína	Energía	Fibra	Precio
Gallinaza	23	22	45	7.60
Sorgo de grano	8	55	25	28.00
Cascarilla de cacahuate	6	29	55	15.00
Pulido de Arroz "A"	14	46	30	40.00
Pulido de Arroz "B"	12	38	40	40.00
Pulido de Arroz "C"	10	35	45	40.00
Paja de Avena "A"	5	30	65	36.00
Paja de Avena "B"	8	30	62	36.00
Melaza	2	73	0	5.50

TABLA 5-2 : REQUERIMIENTOS NUTRICIONALES DIARIOS DE GANADO BOVINO

	Requerimientos nutricionales (Kg.)			
Tipo de animal	Proteína	Energía	Fibra	Número
En destete	1.6	8.0	0.4	15
De engorda	3.6	6.6	19.8	10
En producción (menos de 10 lts.)	4.2	10.8	15.0	5
En producción (más de 10 lts.)	4.8	10.2	15.0	10

Nota: Además de los requisitos nutricionales, se requiere que la dieta de cada animal no contenga más de 80% de gallinaza pues una cantidad mayor podría resultarles nociva.

— o —

Este caso no es muy diferente del problema de asignación de cajeras. De nuevo debe cumplirse una serie de requerimientos a un costo mínimo. Se comienza, como siempre, por definir las variables de decisión del modelo. Se utilizarán cinco grupos de nueve variables cada uno como se muestra a continuación:

Grupo 1		Grupo 2		Grupo 3	
X_1	GalliDes	X_{10}	GalliEng	X_{19}	GalliPrB
X_2	SorgoDes	X_{11}	SorgoEng	X_{20}	SorgoPrB
X_3	CascaDes	X_{12}	CascaEng	X_{21}	CascaPrB
X_4	Pul–ADes	X_{13}	Pul–AEng	X_{22}	Pul–APrB
X_5	Pul–BDes	X_{14}	Pul–BEng	X_{23}	Pul–BPrB
X_6	Pul–CDes	X_{15}	Pul–CEng	X_{24}	Pul–CPrB
X_7	Paj–ADes	X_{16}	Paj–AEng	X_{25}	Paj–APrB
X_8	Paj–BDes	X_{17}	Paj–BEng	X_{26}	Paj–BPrB
X_9	MelazDes	X_{18}	MelazEng	X_{27}	MelazPrB

Grupo 4		Grupo 5	
X_{28}	GalliPrA	X_{37}	Gallinaz
X_{29}	SorgoPrA	X_{38}	Sorgo
X_{30}	CascaPrA	X_{39}	Cascaril
X_{31}	Pul–APrA	X_{40}	PulidAlt
X_{32}	Pul–BPrA	X_{41}	PulidMed
X_{33}	Pul–CPrA	X_{42}	PulidBaj
X_{34}	Paj–APrA	X_{43}	Paja "A"
X_{35}	Paj–BPrA	X_{44}	Paja "B"
X_{36}	MelazPrA	X_{45}	Melaza

Grupo	Variables	Definición
1	X_1 - X_9	Cantidad en Kg. de los distintos productos destinada a animales en destete.
2	X_{10} - X_{18}	Cantidad en Kg. de los distintos productos destinada a animales de engorda.
3	X_{19} - X_{27}	Cantidad en Kg. de los distintos productos destinada a animales en producción baja.
4	X_{28} - X_{36}	Cantidad en Kg. de los distintos productos destinada a animales en producción alta.
5	X_{37} - X_{45}	Cantidad total a comprar de los distintos productos productos en Kg.

Las variables del grupo 5 se definen sencillamente por comodidad, para evitar el tener que manejar una función objetivo muy aparatosa. En concreto, esta se puede escribir simplemente:

$$\text{Min } z = 7.6\ X_{37} + 28\ X_{38} + 15\ X_{39} + 40\ X_{40} + 40\ X_{41} + 40\ X_{42} + 36\ X_{43} + 36\ X_{44} + 5.5\ X_{45}\ .$$

Sin embargo, hay que asegurarse de que estas variables realmente representen los consumos totales de los productos. Esto se hace mediante nueve *restricciones definicionales*, llamadas así porque definen una variable en términos de otras. Cada una de estas restricciones tendrá la siguiente forma:

Compras Totales de un producto	=	Suma de las cantidades requeridas para cada tipo de animal.

Ejemplo: $X_{37} = X_1 + X_{10} + X_{19} + X_{28}$.

Las restricciones correspondientes a la satisfacción de los requerimientos nutricionales de cada animal no presentan ningún problema. Sus lados derechos son las cantidades requeridas de los nutrientes multiplicadas por el número de vacas de cada tipo. Por ejemplo, el requerimiento de proteína de los animales en destete puede escribirse así:

$$0.23X_1 + 0.08X_2 + 0.06X_3 + 0.14X_4 + 0.12X_5 + 0.10X_6 + 0.05X_7 + 0.08X_8 + 0.02X_9 \geq 15 \times 1.6 = 24 .$$

Las restricciones sobre utilización de la gallinaza pueden escribirse según el ejemplo que sigue, correspondiente a animales en destete:

$$\frac{X_1}{X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 + X_8 + X_9} \leq 0.80 ,$$

que equivale a la restricción lineal

$$0.2X_1 - 0.8X_2 - 0.8X_3 - 0.8X_4 - 0.8X_5 - 0.8X_6 - 0.8X_7 - 0.8X_8 - 0.8X_9 \leq 0 .$$

De igual manera se construyen las que corresponden a los consumos de gallinaza de animales de engorda (X_{10}), animales en baja producción (X_{19}) y animales en alta producción (X_{28}). Con esto queda completo el modelo, que se reproduce en las siguientes páginas, acompañado por el correspondiente reporte de resultados.

REPORTE 4

PROBLEMA DE ELABORACION DE DIETA PARA GANADO BOVINO

CASO 4 : CENTRO EXPERIMENTAL AGROPECUARIO "EL PEÑON"

OBJETIVO : MINIMIZAR COSTO TOTAL DE LA RACION DIARIA

SOLUCION OPTIMA ENCONTRADA DESPUES DE

22 ITERACIONES DE FASE I
19 ITERACIONES DE FASE II

DEL METODO SIMPLEX REVISADO.

LOS RESULTADOS NO LISTADOS EN LAS SECCIONES 2-5 VALEN CERO.

1 - VALOR MINIMO DE LA FUNCION OBJETIVO

$ Ración 9694.9949

2 - VALORES OPTIMOS DE LAS VARIABLES DE DECISION

X (1)	GalliDes	92.477064
X (9)	MelazDes	136.51376
X (10)	GalliEng	337.02128
X (12)	CascaEng	84.255319
X (17)	Paj-BEng	.00000000
X (19)	GalliPrB	149.50820
X (20)	SorgoPrB	.00000000
X (21)	CascaPrB	14.038748
X (22)	Pul-APrB	.00000000
X (23)	Pul-BPrB	.00000000

X (24)	Pul-CPrB	.00000000
X (27)	MelazPrB	23.338301
X (28)	GalliPrA	291.14754
X (30)	CascaPrA	34.515648
X (36)	MelazPrA	38.271237
X (37)	Gallinaz	870.15408
X (39)	Cascaril	132.80972
X (43)	Paja "A"	9.65894031E-15
X (45)	Melaza	198.12330

3 - HOLGURA (+) Y EXCESO (-) EN LAS RESTRICCIONES

3.-	- Des-Fibr	35.614679
4.-	- Eng-Prot	46.570213
5.-	- Eng-Ener	32.578723
7.-	- PrB-Prot	14.695976
10.-	- PrA-Prot	21.800298
22.-	+ LimGaDes	90.715596

4 - VALORES DE LAS VARIABLES DUALES

1.-	Des-Prot	26.532110
2.-	Des-Ener	6.8073394
6.-	Eng-Fibr	19.319149
8.-	PrB-Ener	1.5767511
9.-	PrB-Fibr	18.534128
11.-	PrA-Ener	1.5767511
12.-	PrA-Fibr	18.534128
13.-	TotGalli	-7.6000000
14.-	TotSorgo	-9.8497168
15.-	TotCasca	-15.000000
16.-	TotPul-A	-10.634516
17.-	TotPul-M	-12.361788
18.-	TotPul-B	-13.241192
19.-	TotPaj-A	-36.000000
20.-	TotPaj-B	-16.352340
21.-	TotMelaz	-5.5000000
23.-	LimGaEng	-5.4680851
24.-	LimGaPrB	-5.4362146
25.-	LimGaPrA	-5.4362146

5 - COSTOS REDUCIDOS DE LAS VARIABLES DE DECISION

2.-	SorgoDes	3.9831113
3.-	CascaDes	11.433945
4.-	Pul-ADes	3.7886441
5.-	Pul-BDes	6.5911462
6.-	Pul-CDes	8.2054124

7.-	Paj-ADes	32.631193
8.-	Paj-BDes	12.187570
11.-	SorgoEng	.64546152
13.-	Pul-AEng	.46430288
14.-	Pul-BEng	.25966072
15.-	Pul-CEng	.17310714
16.-	Paj-AEng	19.068085
18.-	MelazEng	1.1255319
25.-	Paj-APrB	19.130820
26.-	Paj-BPrB	3.91839427E-02
34.-	Paj-APrA	19.130820
35.-	Paj-BPrA	3.91839427E-02
38.-	Sorgo	18.150283
40.-	Pulido-A	29.365484
41.-	Pulido-B	27.638212
42.-	Pulido-C	26.758808
44.-	Paja "B"	19.647660

6 - RANGOS DE VARIACION DE LOS COSTOS DE LA FUNCION OBJETIVO

VARIABLE		LIMITE INFERIOR	ACTUAL	LIMITE SUPERIOR
X (1)	GalliDes	-5.94247	.000000	6.66068
X (2)	SorgoDes	-3.98311	.000000	Ilimitado
X (3)	CascaDes	-11.4339	.000000	Ilimitado
X (4)	Pul-ADes	-3.78864	.000000	Ilimitado
X (5)	Pul-BDes	-6.59115	.000000	Ilimitado
X (6)	Pul-CDes	-8.20541	.000000	Ilimitado
X (7)	Paj-ADes	-32.6312	.000000	Ilimitado
X (8)	Paj-BDes	-12.1876	.000000	Ilimitado
X (9)	MelazDes	-4.83913	.000000	5.98015
X (10)	GalliEng	-.328865	.000000	4.67273
X (11)	SorgoEng	-.645462	.000000	Ilimitado
X (12)	CascaEng	-3.805052E-02	.000000	.180801
X (13)	Pul-AEng	-.464303	.000000	Ilimitado
X (14)	Pul-BEng	-.259661	.000000	Ilimitado
X (15)	Pul-CEng	-.173107	.000000	Ilimitado
X (16)	Paj-AEng	-19.0681	.000000	Ilimitado
X (17)	Paj-BEng	-19.6477	.000000	3.918394E-02
X (18)	MelazEng	-1.12553	.000000	Ilimitado
X (19)	GalliPrB	1.597192E-14	.000000	1.192659E-14
X (20)	SorgoPrB	-18.1503	.000000	-4.189357E-16
X (21)	CascaPrB	9.673292E-16	.000000	3.709009E-02
X (22)	Pul-APrB	-29.3655	.000000	4.692427E-16
X (23)	Pul-BPrB	-27.6382	.000000	4.692427E-16
X (24)	Pul-CPrB	-26.7588	.000000	4.692427E-16
X (25)	Paj-APrB	-19.1308	.000000	Ilimitado
X (26)	Paj-BPrB	-3.918394E-02	.000000	Ilimitado
X (27)	MelazPrB	7.305246E-16	.000000	26.5735

X (28)	GalliPrA	-1.192659E-14	.000000	-1.597192E-14
X (29)	SorgoPrA	4.189357E-16	.000000	Ilimitado
X (30)	CascaPrA	-5.55205	.000000	-9.673292E-16
X (31)	Pul-APrA	-4.692427E-16	.000000	Ilimitado
X (32)	Pul-BPrA	-4.692427E-16	.000000	Ilimitado
X (33)	Pul-CPrA	-4.692427E-16	.000000	Ilimitado
X (34)	Paj-APrA	-19.1308	.000000	Ilimitado
X (35)	Paj-BPrA	-3.918394E-02	.000000	Ilimitado
X (36)	MelazPrA	-1.12553	.000000	-7.305246E-16
X (37)	Gallinaz	6.39773	7.60000	12.1426
X (38)	Sorgo	9.84972	28.0000	Ilimitado
X (39)	Cascaril	9.44795	15.0000	16.4694
X (40)	Pulido-A	10.6345	40.0000	Ilimitado
X (41)	Pulido-B	12.3618	40.0000	Ilimitado
X (42)	Pulido-C	13.2412	40.0000	Ilimitado
X (43)	Paja "A"	16.9319	36.0000	Ilimitado
X (44)	Paja "B"	16.3523	36.0000	Ilimitado
X (45)	Melaza	4.37447	5.50000	25.2182

7 - RANGOS DE VARIACION DE LOS LADOS DERECHOS

RESTRICCION		LIMITE INFERIOR	ACTUAL	LIMITE SUPERIOR
1.-	Des-Prot	6.27397	24.0000	70.0621
2.-	Des-Ener	41.1064	120.000	767.000
3.-	Des-Fibr	Ilimitado	6.00000	41.6147
4.-	Eng-Prot	Ilimitado	36.0000	82.5702
5.-	Eng-Ener	Ilimitado	66.0000	98.5787
6.-	Eng-Fibr	132.564	198.000	Ilimitado
7.-	PrB-Prot	Ilimitado	21.0000	35.6960
8.-	PrB-Ener	37.3404	54.0000	67.0833
9.-	PrB-Fibr	60.3727	75.0000	108.462
10.-	PrA-Prot	Ilimitado	48.0000	69.8003
11.-	PrA-Ener	74.6809	102.000	134.167
12.-	PrA-Fibr	114.037	150.000	204.872
13.-	TotGalli	Ilimitado	.000000	870.154
14.-	TotSorgo	.000000	.000000	32.4157
15.-	TotCasca	Ilimitado	.000000	132.810
16.-	TotPul-A	.000000	.000000	24.3034
17.-	TotPul-M	.000000	.000000	18.4851
18.-	TotPul-B	.000000	.000000	16.7199
19.-	TotPaj-A	Ilimitado	.000000	9.658940E-15
20.-	TotPaj-B	.000000	.000000	81.8182
21.-	TotMelaz	Ilimitado	.000000	198.123
22.-	LimGaDes	-90.7156	.000000	Ilimitado
23.-	LimGaEng	-219.980	.000000	88.0000
24.-	LimGaPrB	-67.7034	.000000	14.3379
25.-	LimGaPrA	-100.433	.000000	35.2511

Modelo 4 : Centro Agropecuario Experimental El Peñón

Min Z =	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23
R1 Des-Prot	0.23	0.00	0.06	0.14	0.12	0.10	0.05	0.00	0.02														
R2 Des-Ener	0.22	0.55	0.29	0.46	0.30	0.35	0.30	0.30	0.73														
R3 Des-Fibr	0.05	0.25	0.55	0.30	0.40	0.05	0.65	0.62															
R4 Eng-Prot										0.23	0.00	0.06	0.14	0.12	0.10	0.05	0.00	0.02					
R5 Eng-Ener										0.22	0.55	0.29	0.46	0.30	0.35	0.30	0.30	0.73					
R6 Eng-Fibr										0.05	0.25	0.55	0.30	0.40	0.05	0.65	0.62						
R7 PrB-Prot																			0.23	0.00	0.06	0.14	0.12
R8 PrB-Ener																			0.22	0.55	0.29	0.46	0.30
R9 PrB-Fibr																			0.05	0.25	0.55	0.30	0.40
R10 PrA-Prot																							
R11 PrA-Ener																							
R12 PrA-Fibr																							
R13 TotGalli	1.00									1.00									1.00				
R14 TotSorgo		1.00									1.00									1.00			
R15 TotCasca			1.00									1.00									1.00		
R16 TotPul-A				1.00									1.00									1.00	
R17 TotPul-M					1.00									1.00									1.00
R18 TotPul-B						1.00									1.00								
R19 TotPaj-A							1.00									1.00							
R20 TotPaj-B								1.00									1.00						
R21 TotMelaz									1.00									1.00					
R22 LimGaDes	0.20	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00														
R23 LimGaEng										0.20	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.20					
R24 LimGaPrB																			0.20	-0.00	-0.00	-0.00	-0.00
R25 LimGaPrA																							

	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.60	28.00	15.00	40.00	80.00	40.00	36.00	36.00	5.50		
	X24	X25	X26	X27	X28	X29	X30	X31	X32	X33	X34	X35	X36	X37	X38	X39	X40	X41	X42	X43	X44	X45		
R1 Des-Prot																							≥	24.00
R2 Des-Ener																							≥	120.00
R3 Des-Fibr																							≥	6.00
R4 Eng-Prot																							≥	36.00
R5 Eng-Ener																							≥	66.00
R6 Eng-Fibr																							≥	190.00
R7 PrB-Prot	0.10	0.05	0.00	0.02																			≥	21.00
R8 PrB-Ener	0.35	0.30	0.30	0.73																			≥	54.00
R9 PrB-Fibr	0.05	0.65	0.62																				≥	75.00
R10 PrA-Prot					0.23	0.00	0.06	0.14	0.12	0.10	0.05	0.00	0.02										≥	40.00
R11 PrA-Ener					0.22	0.55	0.29	0.46	0.30	0.35	0.30	0.30	0.73										≥	102.00
R12 PrA-Fibr					0.05	0.25	0.55	0.30	0.40	0.05	0.65	0.62											≥	150.00
R13 TotGalli					1.00									-1.00									=	0.00
R14 TotSorgo						1.00									-1.00								=	0.00
R15 TotCasca							1.00									-1.00							=	0.00
R16 TotPul-A								1.00									-1.00						=	0.00
R17 TotPul-M									1.00									-1.00					=	0.00
R18 TotPul-B	1.00									1.00									-1.00				=	0.00
R19 TotPaj-A		1.00									1.00									-1.00			=	0.00
R20 TotPaj-B			1.00									1.00									-1.00		=	0.00
R21 TotMelaz				1.00									1.00									-1.00	=	0.00
R22 LimGaDes																							≤	0.00
R23 LimGaEng																							≤	0.00
R24 LimGaPrB	-0.00	-0.00	-0.00	-0.00																			≤	0.00
R25 LimGaPrA					0.20	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00										≤	0.00

Gracias a la definición de las variables, la sección 2 del reporte incluye dos tipos de datos de gran utilidad para el Sr. Téllez. En primer lugar, los valores de las variables X_1 a X_{36} indican la composición óptima de la ración para cada tipo distinto de animal. Así, por ejemplo, la dieta de las vacas en destete debe elaborarse con 92.5 Kg. de Gallinaza y 136.5 Kg. de Melaza (en la sección 2, $X_1 = 92.47$; X_2 a X_8 no están listadas y por tanto valen cero; $X_9 = 136.51$). Por otro lado, las variables X_{37} a X_{45} indican los totales a comprar de cada producto y, en la sección 1, el valor de la función objetivo indica el costo minimizado de la ración diaria.

La sección 3 de Holguras y Excesos es de vital interés para Téllez, pues le informa cuáles requerimientos nutricionales se están cumpliendo exactamente y cuáles holgadamente y con cuánto exceso. Esta es información que un veterinario debe mantener al día. Los resultados indican, por ejemplo, que en la restricción 4, que corresponde al requerimiento diario de Proteína de Vacas de Engorda, hay un exceso de 46.6 Kg. Como hay 10 animales de este tipo, esto significa que cada uno comería aproximadamente 4.6 Kg. más de lo que necesita de proteína cada día.

La sección de Variables Duales da los precios sombra de las restricciones, esto es, exactamente cuánto cuesta satisfacer los diversos requerimientos nutricionales. Por ejemplo, en el óptimo, cada Kilogramo de Proteína para las vacas en destete (Restricción 1) cuesta 26.53.

Hay aún más información de costos en la sección 5. En concreto, en ella se puede averiguar cuáles son los mejores sustitutos para los productos que actualmente componen la dieta óptima. Por ejemplo, nótese que el Sorgo (X_{38}), a 18.15, tiene el menor costo reducido entre los productos actualmente no comprados. Esto indica que, manteniendo todo lo demás constante, sería la primera opción sustituto si llegara a escasear o encarecerse alguno de los tres productos óptimos; Gallinaza, Cascarilla y Melaza.

Finalmente, obsérvese en la sección 6 que los costos actuales de los productos en la dieta óptima se encuentran todos ellos cercanos a alguno de sus límites de variación, lo que implica que la solución del modelo es moderadamente sensible a cambios en los precios de estos productos.

---

* Este es un caso real, aunque simplificado para propósitos didácticos. Las técnicas modernas para la elaboración de dietas de ganado bovino son bastante más complicadas de lo que aquí se ha descrito, pero aún así, el uso de la PL en este campo ha tenido un éxito extraordinario.

CONCLUSIONES : EL USO PRUDENTE DE LA PROGRAMACION LINEAL

Los capítulos anteriores han pretendido mostrar con ejemplos concretos que los modelos de Programación Lineal son de gran utilidad para analizar muchos problemas de la vida real. El uso de la palabra "analizar", en lugar de "resolver", es deliberado. Se puede resolver --entendiendo por resolver el dar la respuesta correcta-- un ejercicio de texto. Los problemas de la vida real rara vez muestran esta docilidad. En primer lugar, casi nunca hay una sola respuesta satisfactoria. Y, en segundo, muchas veces lo verdaderamente arduo --y que justifica los altos salarios de los tomadores de decisiones-- es definir *cuál* es el problema, porque una vez logrado esto, el análisis puede resultar relativamente sencillo.

Un modelo de PL bien utilizado es una herramienta poderosa y flexible, pero la información que provee es *sólo* uno --*y*, frecuentemente, no el principal-- de los muchos elementos de juicio que intervienen en la toma de una decisión. En realidad es prácticamente imposible construir un modelo matemático que tome en cuenta todos los aspectos de un sistema real, sobre todo, porque lo normal es que no todos sean cuantificables. Un amigo mío, a quien debo el tema de esta tesis, lo ha resumido en una frase muy acertada: *"La realidad es sucia y ruidosa."*

Quiero concluir este trabajo mencionando algunas de las dificultades, no todas ellas de carácter técnico, que pueden presentarse durante la implementación de un modelo de PL en una organización.

En primer lugar, el Modelo Matemático Lineal lleva implícitos una serie de supuestos que el sistema real teóricamente debe cumplir. Estos supuestos son cuatro:

1 - Proporcionalidad: Para cada variable, el consumo del recurso disponible (lado derecho) de cada restricción y la contribución total a la función objetivo deben ser estrictamente proporcionales al valor de la variable. Por ejemplo, si la manufactura de un producto requiere 3 Kg. de acero, dos productos deben requerir 6 Kg., diez productos, 30 Kg., etc. Similarmente, si el costo unitario de producción es $2, el costo de dos productos debe ser $4, de diez productos $20, etc.

2- Aditividad: El consumo total de recursos debe ser la suma de los recursos consumidos individualmente por cada variable. además, el valor total de la función objetivo debe ser la suma de las contribuciones individuales de cada variable.

3- Divisibilidad: Las distintas variables del modelo pueden tomar valores fraccionarios.

4- Determinístico: Los datos del modelo, costos de la función objetivo, lados derechos y coeficientes de las restricciones son todos constantes conocidas.

Es evidente que son muy pocos los sistemas de la vida real que cumplen con los supuestos. Sin embargo, la popularidad actual de la Programación Lineal es una prueba elocuente de que aunque los modelos no sean mas que aproximaciones de los verdaderos problemas, son aproximaciones sumamente útiles.

Un segundo punto es que, ordinariamente, los modelos son formulados por técnicos en PL, no por los tomadores de decisiones que desean utilizarlos. La marcha cotidiana de la organización requiere de la constante atención del gerente o del director. Por eso, aunque tenga la capacidad de diseñar su propio modelo, es lógico que contrate a un especialista. El problema radica en el diferente modo de trabajar y la distinta escala de valores que tienen el tomador de decisiones y el técnico. El primero es normalmente un hombre que yá está íntimamente familiarizado con el sistema y que *solamente* desea una herramienta, *fácil de usar*, que se encargue de hacer las cuentas con rapidez y precisión, dejándole más tiempo libre para emplear su experiencia e intuición en el manejo del sistema, que es para lo que está contratado. Es un hombre cuya principal preocupación son, sencillamente, los centavos. En cambio, el técnico es, por lo general, un consultor sin ninguna experiencia previa del sistema bajo estudio. *Su* principal preocupación es realizar un trabajo serio, intachable desde el punto de vista teórico. Sin embargo, los recursos de tiempo y de dinero requeridos para trabajar de esta manera son elevados, especialmente para la pequeña y mediana industria. En el duro mundo de los negocios, su labor será evaluada desde un punto de vista de estricto costo-beneficio y a corto plazo, pero esto es algo que la mayoría de los técnicos deben aprender en la práctica. La posibilidad de fricciones entre dos mentalidades tan distintas es patente.

El origen de muchas dificultades es simplemente la falta de comunicación entre el técnico y el usuario final del modelo. Con bastante frecuencia, aquél piensa que el éxito del proyecto está garantizado si obtiene el apoyo de los altos jefes de la organización. Pero en realidad, muchas veces es condición necesaria y suficiente para el éxito que el *usuario directo* del modelo esté convencido, pues él será el más autorizado y entusiasta vendedor del proyecto (así como puede ser su más eficaz y encarnizado opositor). Por tanto, en el trabajo de aplicación, una de las cualidades más necesarias en un técnico es la *capacidad de ponerse en los "zapatos de su cliente"*. La mayoría de los fracasos de implementación no son causados por problemas de tecnología sino por problemas de zapatos.

Finalmente, algunas recomendaciones que conviene tener en mente durante la fase de diseño del modelo, son las siguientes:

1 - Una gran parte de los beneficios de la aplicación de la PL a un problema concreto se obtienen de la construcción misma del modelo, más que del cálculo de la solución óptima. El proceso de modelaje obliga a estudiar profunda y ordenadamente los diversos elementos constitutivos del sistema real, lo que aumenta considerablemente la comprensión que se tiene del

mismo. Por eso es sumamente recomendable que el usuario final participe activamente en la formulación del modelo, lo cual, además, facilitará la comunicación antes recomendada y servirá al técnico como un constante punto de referencia.

2 - El modelo no debe ser más complicado de lo necesario. En todo caso, puede crecer después, pero, en principio, mientras más *transparente*, esto es, mientras menos ocultos por las matemáticas estén los procesos del sistema real, mejor. Además, este criterio permite al técnico poner en ejercicio una virtud muy necesaria en la práctica profesional: la sobriedad. Ella le llevará a *armonizar* en su trabajo la actitud de máxima seriedad académica del investigador con la preocupación por los centavos del gerente.

3 - Ningún modelo es perfecto, pero si se descubre que el sistema está muy alejado de los supuestos de linealidad, debe hacerse un juicio del modelo en función de su utilidad. Si los datos que proporciona están tan lejos de la realidad que resultan de poco valor práctico, es posible que deba descartarse la PL como la herramienta de análisis para esta situación. Hay que recordar, sin embargo, que, aún en tal caso, se ha obtenido un beneficio importante del proceso mismo de modelaje. De hecho, el conocimiento adquirido del sistema puede ser tal que se pueda optar sin vacilar por alguna otra de la gran variedad de técnicas de optimización utilizadas hoy en día.

4 - *No hay que perder de vista el sistema real*, esto es, no hay que llegar a pensar (a fuerza de trabajar en él) que el modelo *es* el problema a resolver.

5 - Un modelo no debe forzarse a hacer algo para lo que no fué diseñado. El planteamiento de todo modelo es influenciado no sólo por las peculiaridades del sistema que pretende copiar, o la herramienta técnica disponible (en este caso PL), sino también, de manera decisiva, por sus diseñadores y sus usuarios. Es una herramienta a la medida de una situación muy concreta. Por eso debe actuarse con precaución cuando se pretenda aplicar ese modelo a *otra* situación muy concreta, aunque sea parecida.

6 - Finalmente, una recomendación de orden práctico es cuidar la selección del software que se utilizará para resolver el modelo una vez que haya sido formulado; especialmente, si se decide invertir en un programa comercial. Algunas cualidades obvias de estos paquetes son velocidad, precisión, límite de restricciones que se pueden manejar, que tenga Análisis de Sensibilidad, etc. Sin embargo, hay características menos evidentes que son mucho más apreciadas en el largo plazo. En concreto, un buen paquete debe ser:

a) Interactivo: No sólo el técnico, sino también el tomador de decisiones debe ser capaz de manejar el paquete. Por tanto, el programa mismo debe guiar al usuario, hasta que domine su funcionamiento, mediante textos claros y concisos que aparenzcan en la pantalla, en castellano, por supuesto. Un buen paquete debe poderse utilizar sin consultar ningún manual.

b) Portátil: Esto es, que el paquete pueda correr en una gran variedad de máquinas (en particular, en la que ya se ha adquirido). Entre otras ventajas, esto evita que la salida del mercado de la computadora inicialmente utilizada --un fenómeno frecuente en nuestros días--, implique también la obsolescencia del paquete.

c) Flexible: Es muy frecuente que los modelos lineales sufran ligeras modificaciones a lo largo de su vida útil. Por eso, todos los buenos paquetes de PL poseen rutinas de edición que facilitan la tarea de creación y actualización de modelos.

Cabe mencionar que el Paquete Interactivo de Programación Lineal, descrito detalladamente en el Apéndice II, posee todas estas características, pues sin ellas no cumpliría el objetivo con el que se diseñó.

En resumen, los Modelos de Programación Lineal pueden ayudar a tomar mejores decisiones, pero no pueden hacer más fácil el trabajo de tomar decisiones. El papel del tomador de decisiones --con sus cualidades *irremplazables* de experiencia, intuición y buen juicio-- de ninguna manera se ve reducido: el uso de técnicas como la Programación Lineal amplía considerablemente su campo de trabajo, permitiéndole alcanzar niveles de excelencia previamente insospechados. Su tarea, en efecto, se vuelve más exigente, pero las recompensas que pueden alcanzarse --materiales y no materiales-- son suficiente incentivo para quienes desean poner en su trabajo lo mejor de si mismos.

Referencias Bibliográficas

1. Agin, Norman I. "The Conduct of Operations Research Studies", en *Handbook of Operations Research. Models and Applications. Vol 1*. Editado por Joseph J. Moder y Salah E. Elmaghraby. New York: Van Nostrand Reinhold Company, 1978.

2. Bradley, Stephen P., Arnoldo C. Hax y Thomas L. Magnanti. *Applied Mathematical Programming*. Reading, Mass.: Addison—Wesley, 1977.

3. Buffa, Elwood S. y James S. Dyer. *Essentials of Management Science / Operations Research*. New York: Wiley, 1978.

4. Davis, K. Roscoe y Patrick G. McKeown. *Quantitative Models for Management*. Boston: Kent Publishing Co., 1981.

5. Gass, S. I. *Linear Programming*. New York: McGraw—Hill, 1958.

6. Gillett, Billy E. *Introduction to Operations Research: A Computer—Oriented Algorithmic Approach*. New York: McGraw—Hill, 1976.

7. Hadley, George. *Linear Programming*. Reading, Mass.: Addison—Wesley, 1962.

8. Harvey, Charles M. *Operations Research: An Introduction to Linear Optimization and Decision Analysis*. New York: North Holland, 1979.

9. Henderson, Alexander y Robert Schlaifer. "Mathematical Programming: Better Information for Better Decision Making", en *New Decision Making Tools for Managers*. Editado por Edward C. Bursk. Harvard University Press, 1963.

10. Hillier, Frederick S. y Gerald J. Lieberman. *Introduction to Operations Research. Third Edition*. San Francisco: Holden—Day, 1980.

11. Luenberger, David G. *Introduction to Linear and Nonlinear Programming*. Reading, Mass.: Addison—Wesley, 1973.

12. Phillips, Don T., A. Ravindran y James Solberg. *Operations Research: Principles and Practice*. New York: Wiley, 1976.

13. Shapiro, Roy D. *Optimization Models for Planning and Allocation: Text and Cases in Mathematical Programming*. New York: Wiley, 1984.

14. Taha, Hamdy A. *Operations Research, an Introduction. Third Edition*. New York: Wiley, 1984.

15. Thesen, Arne. *Computer Methods in Operations Research*. New York: Academic Press, 1978.

16. Vatter, Paul A. et al. *Quantitative Methods in Management. Text and Cases*. Homewood, Ill.: Richard D. Irwin, Inc., 1978.

17. Woolsey, Robert E. *Operations Research for Immediate Application: A Quick & Dirty Manual*. New York: Harper & Row, 1975.

Lecturas Recomendadas

El libro de Taha —la tercera edición— tiene una muy clara y completa exposición del tema de Programación Lineal con un nivel matemático semejante al del Apéndice I. Es una compra muy recomendable para el lector interesado, pués es además un excelente libro de Investigación de Operaciones. Otra opción introductoria interesante es el texto de Gillett, que incluye programas sencillos que pueden ser corridos en microcomputadoras.

Un desarrollo más riguroso de la teoría matemática de la Programació Lineal puede encontrarse en el texto de Hadley, que es un auténtico clásico.

Por último, Shapiro, utilizando el Método del Caso, presenta en su libro una gran variedad de aplicaciones reales de P.L. y otras técnicas de optimización, enfatizando el desarrollo de modelos y la interpretación de resultados. El lector podrá notar la enorme influencia que este autor ha tenido en el desarrollo de este manual.

Apéndice I : El Algebra del Método Simplex.

El Método Simplex es un algoritmo iterativo para la solución de modelos lineales que, mediante la solución de una serie de sistemas de ecuaciones lineales, determina sucesivamente los vértices de la región de soluciones factibles, en forma muy parecida al procedimiento geométrico del capítulo dos. Además, este cálculo se hace de tal manera que para un problema de maximización (minimización) no se evaluará un vértice si este lleva a un valor menor (mayor) de la función objetivo. Así, siguiendo una estrategia de aumentos (reducciones) sucesivas en la función objetivo, se evalúa sólo una fracción del total de vértices de la región de soluciones y puede llegarse rápidamente al óptimo. En este apéndice se expone, a grandes rasgos, el principio algebráico del método, resolviendo numéricamente el Modelo 1, que se muestra a continuación:

Max $z = 1,000\ x + 1,500\ y$ Utilidad

sujeta a

$0.8\ x + 0.4\ y \leq 24$ (1) Chocolate

$0.2\ x + 0.5\ y \leq 10$ (2) Jarabe

$0.1\ y \leq 1.5$ (3) Crema

$x,\ y \geq 0$ No negatividad

El primer paso para resolver este modelo utilizando el Método Simplex es transformarlo de su formato actual a la *Forma Estándar*, que tiene las siguientes características:

1) Todas las restricciones son de igualdad.

2) En cada restricción hay por lo menos una variable cuyo coeficiente es 1.

3) Esta variable no aparece en ninguna otra restricción.

La técnica que se utiliza para transformar las desigualdades en ecuaciones consiste en añadirles una *Variable de Holgura*. Tómese, por ejemplo, la primera restricción:

$$0.8\, x + 0.4\, y \leq 24 .$$

El valor del lado izquierdo de esta restricción nunca será mayor que 24 pero puede ser menor. Puede reemplazarse esta desigualdad con la siguiente ecuación:

$$0.8\, x + 0.4\, y + h_1 = 24 ,$$

donde h_1 es una variable de holgura que se distingue por:

a) Depende totalmente de las dos variables de decisión, pues, como su nombre lo indica, únicamente toma como valor "lo que falte" a ($0.8\, x + 0.4\, y$) para llegar a 24.

b) Como cualquier otra variable en Programación Lineal, h_1 está limitada a valores no negativos. Esto es imprescindible, pués si h_1 tomara un valor negativo, como -1, entonces, para que se cumpliera la ecuación, $(0.8\, x + 0.4\, y)$ tendría que valer 25 y la restricción original no se cumpliría. En cambio, al exigir que h_1 no tome valores negativos, se está garantizando que la restricción original se cumplirá y se puede trabajar con una ecuación, que algebráicamente es mucho más sencilla de manejar que una desigualdad.

Agregando una variable de holgura a cada restricción del modelo y llamando z a la función objetivo, se obtiene la forma estándar del modelo:

$$
\left.
\begin{array}{lr}
0.8\,x + 0.4\,y + h_1 = 24 & (1) \\
0.2\,x + 0.5\,y + h_2 = 10 & (2) \\
0.1\,y + h_3 = 1.5 & (3) \\
1,000\,x + 1,500\,y = z & \text{F.O.}
\end{array}
\right\} \quad [1]
$$

$$x,\ y,\ h_1,\ h_2,\ h_3 \geq 0$$

Obsérvese que el sistema [1] cumple con las condiciones exigidas para la forma estándar; todas las restricciones son ecuaciones, y existe en cada ecuación una variable con coeficiente +1 y que aparece únicamente en esa ecuación. Cualquier variable que cumpla con este requisito se denomina una *Variable Básica*, y el conjunto de variables básicas de un problema en un momento dado se conoce como la *Base*. En el sistema [1], el conjunto { h_1, h_2, h_3 } constituye la base actual. El número de variables básicas nunca puede exceder el número de restricciones del problema. Las variables que no están actualmente en la base se denominan no básicas.

La utilidad de la forma estándar estriba en que una solución factible puede leerse directamente del sistema. Si se reacomodan términos en cada una de las ecuaciones, para que las variables básicas queden todas del mismo lado, se obtiene el sistema [2]:

$$
\left.
\begin{array}{lcll}
h_1 & = & 24 - 0.8\,x - 0.4\,y & (1) \\
h_2 & = & 10 - 0.2\,x - 0.5\,y & (2) \\
h_3 & = & 1.5 - 0.1\,y & (3) \\
z & = & 1{,}000\,x + 1{,}500\,y & \text{F.O.}
\end{array}
\right. \qquad [2]
$$

Nótese como una solución factible puede ser generada de inmediato haciendo las variables x y y, esto es, *las variables no básicas*, iguales a cero. Dicha solución es:

$$
\begin{array}{lcl@{\qquad\qquad}lcl}
h_1 & = & 24 & x & = & 0 \\
h_2 & = & 10 & y & = & 0 \\
h_3 & = & 1.5 & z & = & 0
\end{array}
$$

Como todos estos valores son no negativos, esta es efectivamente una solución factible del sistema —al ser no negativas las holguras se satisfacen automáticamente las restricciones del modelo—, y servirá como punto de arranque para el Método Simplex. Una solución factible obtenida por este procedimiento (a partir de un conjunto de ecuaciones en forma estándar) se denomina una solución factible básica. En toda solución factible básica, las variables no básicas valen cero. De ordinario, aunque no forzosamente, las variables básicas serán mayores que cero. Se puede demostrar que *cada solución básica factible corresponde a uno y sólo ún vèrtice de la regiòn de soluciones factibles.* Como se sabe que alguno de los vértices es óptimo, se tiene la certeza de que alguna de las soluciones básicas factibles es óptima.

El Método Simplex funciona moviéndose de vértice a vértice, comenzando por el que corresponde a la solución básica factible inicial. Estos movimientos se logran transformando el sistema de ecuaciones del vértice actual en otro sistema equivalente, del cual se pueda leer una nueva solución básica factible. La elegancia y eficiencia del Método Simplex radica en el hecho de que nunca permitirá la evaluación de una solución que produzca un valor de función objetivo peor que el actual. Este método de evaluar sólo vértices que resulten beneficiosos generalmente lleva a una solución óptima mucho más rápidamente que una estrategia de evaluación de todos los vértices de la región de soluciones.

La médula del Método, por tanto, es esta transformación que permite cambiar de un vértice a otro, y consiste en hacer que una de las variables básicas abandone la base (asumiendo un valor de cero) para que una de las variables no básicas tome su lugar y pueda tomar así un valor positivo. Para hacer esto, primero debe elegirse la variable que va a entrar a la base. Obsérvense los coeficientes de la función objetivo en el sistema [2]. Por cada unidad que aumente el valor de x, el valor de z crecerá en 1,000, mientras que por cada unidad que aumente y, se incrementará en

1,500. Por tanto, si aumenta el valor de x o y, que actualmente es cero, el valor de la función objetivo aumentará también. Hay varios criterios para decidir cuál de las variables no básicas es la que debe entrar a la base. El mas difundido consiste en elegir aquella que tenga el *mayor coeficiente* en la función objetivo. Por tanto, en el ejemplo, se selecciona para entrar a la base a la variable y.

Como la solución no puede tener más variables básicas que el número de restricciones del problema, el meter y a la base implica sacar h_1, h_2, o h_3. El criterio para decidir cuál es la variable que sale de la base implica determinar cuánto puede incrementarse y sin que se vuelva negativa alguna de las variables del sistema, esto es, *sin perder la factibilidad*. En la primera ecuación de [2] observe que se puede aumentar y hasta 60 sin que h_1 se haga negativa. En la segunda se puede aumentar y hasta 20. En la tercera, y puede aumentar hasta 15. Por tanto esta última es la más restrictiva de las ecuaciones. Cuando y se incrementa a 15, h_3 se vuelve cero y sale de la base, tomando y su lugar.

Formalmente, el criterio que se utiliza para determinar cuál es la variable que debe abandonar la base es el siguiente: Para cada ecuación donde el coeficiente de la variable por entrar es *negativo*, divídase el valor actual de la variable básica de esa ecuación entre el valor absoluto de dicho coeficiente. De estos resultados, el *más cercano a cero* señala la ecuación más restrictiva, pues la variable básica de esta ecuación será la primera en volverse negativa al aumentar el valor de la variable por entrar. En el ejemplo, la variable que debe salir es h_3, como puede verse a continuación:

Variables Básicas		Valor Actual Var. Básicas				Variable por entrar	Cocientes
h_1	=	24	—	$0.8\ x$	—	$0.4\ y$	24 / 0.4 = 60
h_2	=	10	—	$0.2\ x$	—	$0.5\ y$	10 / 0.5 = 20
h_3	=	1.5			—	$0.1\ y$	1.5 / 0.1 = 15
z	=			$1,000\ x$	+	$1,500\ y$	

Una vez que se sabe cuál es la variable que va a entrar a la base y cuál es la que va a salir, debe transformarse el sistema para realizar el intercambio. Esto puede hacerse de la siguiente manera: Primeramente, se toma la ecuación mas restrictiva, en este caso la tercera, y se reacomodan términos para dejar y del lado izquierdo, como corresponde a una variable básica, y h_3 del lado derecho, entre las variables no básicas. Esto es:

$$0.1\ y = 1.5 - h_3 .$$

Despues, se dividen ambos lados entre 0.1 para que el coeficiente de y sea +1, como corresponde a una variable básica en la forma estándar. Esto da por resultado la nueva ecuación:

$$y = 15 - 10\ h_3 .$$

Por último, se sustituye $(15 - 10\,h_3)$ en lugar de y en todas las demas ecuaciones del sistema, incluyendo la función objetivo:

$$
\begin{aligned}
h_1 &= 24 - 0.8\,x - 0.4\,(15 - 10\,h_3) \\
h_2 &= 10 - 0.2\,x - 0.5\,(15 - 10\,h_3) \\
y &= 15 - 10\,h_3 \\
z &= 1{,}000\,x + 1{,}500\,(15 - 10\,h_3)
\end{aligned}
$$

Simplificando este sistema se alcanza una nueva forma estándar, y una nueva solución factible básica:

$$
\left.
\begin{aligned}
h_1 &= 18 - 0.8\,x + 4\,h_3 && (1) \\
h_2 &= 2.5 - 0.2\,x + 5\,h_3 && (2) \\
y &= 15 - 10\,h_3 && (3) \\
z &= 22{,}500 + 1{,}000\,x - 15{,}000\,h_3 && \text{F.O.}
\end{aligned}
\right\} \quad [3]
$$

La nueva base es $\{h_1, h_2, y\}$, y la correspondiente solución básica factible del sistema es:

$$
\begin{aligned}
h_1 &= 18 & \qquad x &= 0 \\
h_2 &= 2.5 & \qquad h_3 &= 0 \\
y &= 15 & \qquad z &= 22{,}500
\end{aligned}
$$

Nótese que esta solución del problema, $x = 0$, $y = 15$, es el equivalente númerico del vértice B de la Gráfica 1 (pag. 16). Se ha completado la primera iteración del Método Simplex, obteniendo un sistema equivalente en forma estándar del cual puede leerse directamente la nueva solución, la cual ha mejorado el valor de la función objetivo, de 0 a 22,500.

Para comenzar la segunda iteración, debe averiguarse si el introducir alguna de las variables no básicas a la base sería beneficioso para la función objetivo. Obsérvense los coeficientes de las variables no básicas en la función objetivo del sistema [4]. El aumentar x en una unidad sumaría $1,000 al valor de z. En cambio, el aumentar h_3 en la misma cantidad le restaría 15,000 a la función objetivo. Por tanto, x es la variable que entrará a la base en esta iteración.

Ahora debe determinarse cuál es la ecuación más restrictiva para aumentos en la variable x. Utilizando nuevamente la prueba de los cocientes se encuentra que en la ecuación (1), x puede crecer hasta 22.5, antes de que h_1 se vuelva negativa, y en la ecuación (2), puede crecer hasta 12.5 antes de que h_2 tome valores negativos. El coeficiente de x en la ecuación (3) no es negativo sino cero, por lo que este cociente no es relevante.

Variables Básicas		Valor Actual Var. Básicas	Variable por entrar			Cocientes
h_1	=	18	$-\ 0.8\ x$	+	$4\ h_3$	18/0.8 = 22.5
h_2	=	2.5	$-\ 0.2\ x$	+	$5\ h_3$	2.5/0.2 = 12.5
y	=	15		−	$10\ h_3$	No relevante
z	=	22,500	$+\ 1,000\ x$	−	$15,000\ h_3$	

Por tanto, la ecuación (2) es la mas restrictiva y h_2 es la variable que sale de la base. Dividiendo ambos lados de la ecuación entre 0.2, y moviendo x al lado izquierdo y h_2 al derecho se obtiene la nueva ecuación:

$$x = 12.5 - 5\,h_2 + 25\,h_3$$

Sustituyendo ($12.5 - 5\,h_2 + 25\,h_3$) en lugar de x en las ecuaciones (1), (3) y en la función objetivo, y simplificando el sistema se llega a la nueva forma estándar:

$$\begin{array}{rclllr} h_1 & = & 8 & +\ 4\,h_2 & -\ 16\,h_3 & (1) \\ x & = & 12.5 & -\ 5\,h_2 & +\ 25\,h_3 & (2) \\ y & = & 15 & & -\ 10\,h_3 & (3) \\ z & = & 35{,}000 & -\ 5{,}000\,h_2 & +\ 10{,}000\,h_3 & \text{F.O.} \end{array} \qquad [4]$$

La base ahora es { h_1, x, y }, y la nueva solución básica factible del sistema es:

$$\begin{array}{rclcrcl} h_1 & = & 8 & \qquad & h_2 & = & 0 \\ x & = & 12.5 & & h_3 & = & 0 \\ y & = & 15 & & z & = & 35{,}000 \end{array}$$

Esta corresponde al vértice C de la región de soluciones factibles, como puede verse en la Gráfica 1. De hecho, el método está siguiendo el mismo camino que se recorrió geométricamente en el capítulo dos, de donde se deduce que la próxima iteración alcanzará el óptimo.

En efecto, los coeficientes de la función objetivo indican que se puede realizar otra iteración, pues aumentar el valor de h_3 en una unidad hará que z aumente en 10,000. Por tanto, h_3 es la variable que entrará a la base. Calculando los cocientes se encuentra que en la ecuación (1), h_3 puede aumentar hasta 0.5, y en la ecuación (3), puede crecer hasta 1.5. En la ecuación (2), el coeficiente de h_3 es positivo y, por tanto, no relevante:

Variables Básicas		Valor Actual Var. Básicas				Variable por entrar	Cocientes
h_1	=	8	+	$4\ h_2$	−	$16\ h_3$	8 / 16 = 0.5
x	=	12.5	−	$5\ h_2$	+	$25\ h_3$	No relevante
y	=	15			−	$10\ h_3$	15 / 10 = 1.5
z	=	35,000	−	$5{,}000\ h_2$	+	$10{,}000\ h_3$	

Realizando las operaciones necesarias se llega al siguiente y último sistema:

$$
\begin{array}{rcrcrcrl}
h_3 & = & 0.5 & - & 0.0625\ h_1 & + & 0.25\ h_2 & (1) \\
x & = & 25 & - & 1.5625\ h_1 & + & 1.25\ h_2 & (2) \\
y & = & 10 & + & 0.625\ h_1 & - & 2.5\ h_2 & (3) \\
z & = & 40{,}000 & - & 625\ h_1 & - & 2{,}500\ h_2 & \text{F.O.}
\end{array}
\qquad [5]
$$

$$
\begin{array}{rclcrcl}
h_3 & = & 0.5 & \qquad & h_2 & = & 0 \\
x & = & 25 & & h_1 & = & 0 \\
y & = & 10 & & z & = & 40{,}000
\end{array}
$$

Nótese que los coeficientes de las variables h_1 y h_2 en la función objetivo son ambos negativos. El introducir a la base cualquiera de estas dos variables causaría una reducción en el valor de la función objetivo. En otras palabras, cualquier otra solución básica reportará un menor valor de z. Por consiguiente, se ha alcanzado algebráicamente la misma solución óptima que geométricamente se obtuvo en el capítulo dos.

* Se han omitido varios aspectos importantes del método, por estar fuera del alcance y objetivos de este texto. En la sección de Referencias Bibliográficas se ha incluido una pequeña lista de lecturas, adecuadas para profundizar en el tema.

Apéndice II : El Paquete Interactivo de Programación Lineal.

Descripción

El Paquete Interactivo de Programación Lineal es un programa escrito en lenguaje Microsoft FORTRAN 77 para crear, editar y resolver modelos de Programación Lineal en microcomputadoras IBM-PC y máquinas compatibles.

Objetivo

A pesar de que en México hay cada vez más usuarios de Programación Lineal —en el sector público, en el medio industrial y empresarial y en el medio universitario— y de que, en los últimos años, las microcomputadoras han tenido una rápida difusión, el *software* comercial disponible ha representado un obstáculo importante para muchos de estos nuevos usuarios. Los paquetes para máquinas micro, como LINDO o MILP-88, no son lo bastante interactivos ("amistosos" sería un mejor término) para quien está dando sus primeros pasos en este campo. En concreto, una opción "Help" no sirve de mucho, si lo único que de ella se obtiene es un mensaje muy corto, en un inglés muy técnico.

Se necesita un nuevo paquete, que aproveche la capacidad de memoria, velocidad, precisión y sobre todo la popularidad del estandar IBM-PC. Un programa lo bastante completo como para satisfacer a un usuario experimentado, y tan interactivo, que el principiante, con un conocimiento básico de PL, pueda usarlo de inmediato. P.I.P.L. es una respuesta concreta a esta necesidad.

Características

Métodos de Solución: P.I.P.L. utiliza el Método Simplex Revisado como algoritmo principal de solución. Con ello se obtienen de inmediato todas las secciones del reporte de resultados salvo los rangos de variación de los costos y de los lados derechos, que se calculan en dos rutinas auxiliares. Una vez resuelto el problema, puede utilizarse el Método Dual Simplex, o repetir el Simplex Revisado, para actualizar la solución tras hacer modificaciones al modelo con las rutinas de Análisis de Sensibilidad.

Capacidad: Se han desarrollado dos versiones del paquete. La primera, diseñada para operar en máquinas con 256 Kb. de memoria principal, puede manejar problemas de hasta 40 restricciones y 40 variables. La segunda versión requiere 512 Kb. y acepta modelos de 100 x 100. En ambos casos, el límite de variables se refiere a las variables de decisión exclusivamente, pues se ha previsto el espacio de memoria que requerirán las variables de holgura y artificiales.

Velocidad: El tiempo requerido para resolver en computadora los modelos lineales depende ante todo del número de restricciones de los mismos, más que del número de variables. Para modelos con no más de quince restricciones, el tiempo promedio de solución de P.I.P.L. es menos de un minuto. Este tiempo crece rápidamente al aumentar el número de restricciones. Por ejemplo, un problema de 45 restricciones y 70 variables fué resuelto, después de 130 iteraciones, en 23 minutos. Por otro lado, hay que hacer notar que el tiempo de solución se reduce aproximadamente a la mitad cuando se utiliza el paquete en una máquina IBM-PC AT, o en una equipada con el coprocesador numérico 8087.

Precisión: El Método Simplex utiliza para cada cambio de base el procedimiento de Gauss-Jordan que es muy veloz, pero puede perder precisión después de muchas iteraciones en matrices grandes. Previendo esto, se incluye en el paquete la opción de reinvertir la base, partiendo de los vectores columna originales, después de un número de iteraciones determinado por el usuario. Esto, por supuesto, alarga mucho el tiempo de solución, lo cual se advierte al usuario mediante un mensaje por pantalla. Un caso concreto es el modelo de 45 restricciones mencionado antes: Reinvirtiendo una vez, después de la iteración 90, el tiempo de solución fue aproximadamente de seis horas y los resultados obtenidos fueron prácticamente idénticos a los anteriores.

Facilidades para el usuario: Desde el principio se procuró diseñar el paquete con una mentalidad de servicio para con el usuario. Concretamente:

a) El paquete funciona por medio de menús, lo que evita el tener que aprender instrucciones o comandos. De hecho, no se requiere ningún manual para manejar P.I.P.L. El usuario que no está familiarizado con el paquete puede pedir que aparezcan por pantalla instrucciones que le orienten en el funcionamiento de las diversas rutinas. En cambio, el técnico en PL puede guiarse exclusivamente por los menús y así trabajar con mayor rapidez.

b) Se pueden poner nombres de hasta ocho caracteres alfanuméricos a las restricciones y las variables para facilitar la lectura de los reportes de resultados.

c) El paquete define y utiliza las variables de holgura y artificiales necesarias en forma totalmente automática, librando al usuario de esta preocupación.

d) Una vez creados, los modelos pueden grabarse en diskette para evitar en el futuro la engorrosa tarea de teclearlos de nuevo. Si además el problema ya ha sido resuelto, la solución óptima también se guarda, con lo cual no tendrá que recalcularse la siguiente vez que se de acceso a ese modelo.

e) Existe la opción de imprimir los reportes de resultados en un archivo de diskette, lo cual permite incorporarlos después, mediante un procesador de textos, en informes u otros documentos.

Listado de rutinas y sus funciones

Para propósitos de exposición se ha dividido el listado del programa en siete bloques que se detallan a continuación:

Bloque: CONTROL

Función: Desplegar los menús y llamar a los demás bloques.

Subrutinas	Descripción
MAIN	Programa principal. LLama a los demás bloques.
MENUS	Despliega los menús de uso más frecuente.

Bloque: **ENTRADA**

Función: Lectura de los datos del modelo lineal.

Subrutinas	Descripción
ACCESO	Lectura de datos por el teclado.
RESTR	Lectura por el teclado de una restricción.
LEEDISC	Lectura de modelos grabados en diskette.

Bloque: **EDICION**

Función: Realizar modificaciones al modelo en memoria.

Subrutinas	Descripción
EDITOR	Despliega el menú y llama a las demás rutinas.
EDITL	Controla listados de restricciones y función objetivo.
LISTR	Lista las restricciones del modelo.
LISTV	Lista los coeficientes de una restricción.
FUNOBJ	Lista los coeficientes de la función objetivo.
EDIT3	Modifica los coeficientes de aluna restricción.
EDIT4	Modifica el lado derecho de alguna restricción.
EDIT5	Modifica los costos de la función objetivo.
EDIT6	Modifica el tipo de alguna restricción ($\leq$, $\geq$ ó =).
EDIT7	Cambia el sentido de la optimización.
EDIT8	Permite agregar restricciones al modelo.
EDIT9	Permite agregar variables al modelo.
EDIT10	Elimina una restricción del modelo.
EDIT11	Elimina una variable del modelo.
EDIT12	Modifica el encabezado y los nombres de variables, restricciones, función objetivo y modelo lineal.

Bloque: **ALGORITMOS**

Función: Agrupa las rutinas para los algoritmos de solucíon; métodos Simplex Revisado, Dual Simplex, reinversión de matrices para garantizar la precisión, y cálculo de los rangos de variación de costos y lados derechos.

Subrutinas	Descripción
OPTIMO	Dirige la resolución del modelo por el M.S.R.
VARHOL	Define las variables de holgura y artificiales.
VARBAS	Construye la solución básica inicial.
VARNOB	Construye el conjunto de variables no básicas.
PRECIS	Define parámetros de precisión numérica.
SIMREV	Algoritmo Simplex Revisado.
SRZJCJ	Cálculo de las $Z_j - C_j$.
ITERA	Algoritmo de Gauss-Jordan para cambiar de base.
RANGOB	Determina rangos de variación de los lados derechos.
RANGOC	Determina rangos de variación de los costos.
DUALSI	Algoritmo Dual Simplex (para Análisis Post-Optimo).
BASE	Construye la base actual y llama a la rutina SLINEQ.
SLINEQ	Método de inversión de matrices por descomposición triangular con la variante de Cholesky.*

---

* Esta subrutina fué originalmente una contribución del Ing. Humberto Cortez Casillas. Ha sido ligeramente simplificada para la función de reinversión que cumple en P.I.P.L. Aprovecho este espacio para expresar mi reconocimiento al Ing. Cortez por su valiosa aportación.

Bloque: **ANALISIS**

Función: Acepta modificaciones al modelo resuelto y calcula de inmediato la nueva solución óptima mediante el Simplex Revisado o el Dual Simplex.

Subrutinas	Descripción
ADESEN	Despliega el menú y llama a las demás rutinas.
ADSEN1	Cambios en los lados derechos de las restricciones.
ADSEN2	Cambios en los costos de la función objetivo.
ADSEN3	Agregar una variable al modelo.
ADSEN4	Agregar una restricción al modelo.

Bloque: **SALIDA**

Función: Imprimir reportes y guardar en diskette el modelo.

Subrutinas	Descripción
REPORT	Programa principal. LLama a las demás rutinas.
UNIDAD	Elige la unidad de salida de resultados (pantalla, impresora o diskette).
MENURES	Elige las secciones de resultados a incluir en el reporte.
LISTPL	Rutina de listado del modelo lineal completo.
IMPRIM	Rutina de impresión de reportes.
GUARDA	Graba los modelos en un archivo de diskette.

Bloque: **AUXILIAR**

Función: Agrupa diversas rutinas de uso frecuente.

Subrutinas	Descripción
ALTO	Detiene la ejecución del programa.
COSTOS	Invierte el signo de los costos (para minimizar).
INIABC	Inicializa las matrices A, BI o el vector B.
INICIA	Inicializa las variables del paquete.
MINIMO	Calcula el mínimo de un vector real.
ORDEN	Ordena, sin modificarlo, un vector de números enteros.
PAUSA	Detiene la ejecución del programa y pide por pantalla teclear un RETURN para continuar.
SELECT	Activa y desactiva las diversas secciones del reporte de resultados.
TEXTOS	Despliega los mensajes para el usuario, grabados en diskette, en la pantalla.
VITICS	Limpia pantalla y/o salta renglones.*

* Contribución original de Ignacio Vizcaino Tapia.

Listado completo de P.I.P.L.

```
$debug
$large
$storage:2

C     PAQUETE INTERACTIVO DE PROGRAMACION LINEAL
C
C
C     Versión de 512 Kb
C
C     Capacidad máxima:   100 restricciones
C                         100 variables de decisión.
C
C
C     Variables principales de PIPL :
C     ------------------------------
C
C     Vectores:
C     ---------
C
C    Nombre      Dimensión    Tipo       Función que realiza
C
C     A         (102, 100)    DP    Matriz de restricciones
C     AUX       (102, 102)    DP    Matriz auxiliar para reinversiones
C     B              (102)    DP    Vector de lados derechos
C     BI        (102, 102)    DP    Inversa de la Base
C     IBAS           (102)    INT   Vector de indices de variables basicas
C     INBAS          (102)    INT   Vector de indices de var. no basicas
C     IHOL        (202, 2)    INT   Vector de var. de holgura y artificiales
C     MRESTR         (102)    INT   Vector de tipos de restricciones
C     RBSUP          (100)    DP    Cotas sup. de variacion Lados Derechos
C     RBINF          (100)    DP    Cotas inf. de variacion Lados Derechos
C     RCSUP          (102)    DP    Cotas sup. de variacion de los Costos
C     RCINF          (102)    DP    Cotas inf. de variacion de los Costos
C     T1             (102)    DP    Vector para calculos numericos
C     T2 = YJ        (102)    DP    Vector para calculos numericos
C     U           (102, 3)    DP    Vector para calc. num. (reinversion)
C     XB             (102)    DP    Valores de las variables basicas
C     ZJCJ           (102)    DP    Vector de Zj - Cj de var. no basicas
C
C
C     Variables
C     Enteras:    MO       INT      Número original de restricciones
C     ---------   M        INT      Restricciones despues de la Fase I
C                 NO       INT      Número original de variables
C                 MB       INT      Dimension de la base
C                 MAXMIN   INT      +1 = maximizar,   -1 = minimizar
C                 NOBAS    INT      Número de variables no básicas
C                 NH       INT      Número de variables de holgura
C                 NA       INT      Número de variables artificiales
C                 NOH      INT      NO + NH
C                 NOHA     INT      NO + NH + NA
```

```
C       Programa principal de control
        IMPLICIT LOGICAL (S)
        DOUBLE PRECISION A, B
        COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
        COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
        COMMON /FORM/ IMP(8), LUN, FMT
        COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
        CHARACTER IMP, RESP
        CHARACTER*8 FMT, MODELO
        CHARACTER*12 STATUS
        FMT = '        '
        CALL INICIA
        CALL SELECT ('*', '*', '*', '*', '*', '*', '*', ' ')
        LUN = 1
        SMENU = .FALSE.
        WRITE (*, 1000)
        READ (*, '(A1)') RESP
        CALL VITICS (0)
        IF (RESP.EQ.'S'.OR.RESP.EQ.'s'.OR.RESP.EQ.' ') THEN
                SINSTR = .TRUE.
                SMENU = .TRUE.
                CALL VITICS (0)
                WRITE (*, 1006)
                CALL ALTO (LIN)
                GO TO 10
        ENDIF
        CALL VITICS (0)
        WRITE (*, 1001)
        CALL PAUSA
        CALL ALTO (LIN)
        CALL VITICS (0)
        WRITE (*, 1006)
        CALL ALTO (LIN)
        CALL TEXTOS ('MAIN2.TXT   ', 0)
        WRITE (*, 1002)
        READ (*, '(A1)') RESP
        IF (RESP.EQ.'N'.OR.RESP.EQ.'n') THEN
                SINSTR = .TRUE.
                GO TO 10
        ELSE
                SINSTR = .FALSE.
        ENDIF
        CALL TEXTOS ('MAIN3.TXT   ', 0)
        CALL PAUSA
        CALL ALTO (LIN)
        CALL TEXTOS ('MAIN4.TXT   ', 0)
        CALL PAUSA
        CALL ALTO (LIN)
10      IF (SOLOPT) THEN
                STATUS = ' Optima     '
        ELSEIF (SNACOT) THEN
                STATUS = ' No Acotada '
        ELSEIF (SNFACT) THEN
                STATUS = ' No Existe  '
        ELSE
```

```
                STATUS = 'Por Calcular'
        ENDIF
        IF (MO.EQ.O.AND.NO.EQ.O) MODELO = '
        IF (MODELO.EQ.'           ') STATUS = '
        IF (SMENU.AND.SINSTR) THEN
                CALL MENUS (1)
        ELSE
                CALL MENUS (2)
        ENDIF
        SMENU = .TRUE.
        WRITE (*, 1003)
        READ (*, '(I1)', ERR = 10) IRESP
        CALL VITICS (O)
        IF (IRESP.LT.O.OR.IRESP.GT.8) GO TO 10
        IF (IRESP.EQ.O) SMENU = .FALSE.
        IF (IRESP.EQ.1) CALL ACCESO
        IF (MODELO.EQ.'           '.AND.RESP.NE.'1'.AND.RESP.NE.'7'
     *     .AND.RESP.NE.'8') THEN
                WRITE (*, 1004)
                CALL PAUSA
                CALL ALTO (LIN)
                GO TO 10
        ENDIF
        IF (RESP.EQ.'2') CALL EDITOR
        IF (RESP.EQ.'3'.AND.(.NOT.SOLOPT).AND.(.NOT.SNFACT)
     *   .AND.(.NOT.SNACOT)) THEN
                CALL COSTOS
                CALL OPTIMO
                CALL COSTOS
        ENDIF
        IF (RESP.EQ.'4') CALL REPORT
        IF (RESP.EQ.'5'.AND.(SOLOPT.OR.SNFACT.OR.SNACOT)) THEN
                CALL COSTOS
                CALL ADESEN
                CALL COSTOS
        ENDIF
        IF (RESP.EQ.'6') CALL GUARDA
        IF (RESP.EQ.'8') SINSTR = .NOT. SINSTR
        IF (RESP.NE.'7') GO TO 10
        IF (.NOT.SINSTR) THEN
                CALL TEXTOS ('SALIDA.TXT  ', O)
        ELSE
                CALL VITICS (5)
        ENDIF
        WRITE (*, 1005)
        READ (*, '(A1)') RESP
        IF (RESP.EQ.'S'.OR.RESP.EQ.'s') GO TO 10

1000    FORMAT (///////
     * /' ',12X,'( Est s familiarizado con el uso de este paquete'
     * /'O',12X,'[ S/N ]  ?  '\)
1001    FORMAT (
     */'O',12X,'IMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM;'
     */' ',12X,':                                                  :'
     */' ',12X,':                                                  :'
```

```
     */'  ',12X,':        \\\\\\\\\\    \\\\\    \\\\\\\\\\    \               :'
     */'  ',12X,':        [\\\\\\\\[      [      [\\\\\\\\[    [               :'
     */'  ',12X,':        [              \\[\\    [             [\\\\\\\\      :'
     */'  ',12X,':                                                              :'
     */'  ',12X,':                                                              :'
     */'  ',12X,':       PAQUETE INTERACTIVO DE PROGRAMACION LINEAL             :'
     */'  ',12X,':                                                              :'
     */'  ',12X,':                                                              :'
     */'  ',12X,':                                                              :'
     */'  ',12X,':           Though this be madness, yet there is               :'
     */'  ',12X,54H:         method in't.                                        :
     */'  ',12X,':                               Shakespeare - Hamlet           :'
     */'  ',12X,':                                                              :'
     */'  ',12X,':                                                              :'/
     *'  ',12X,'HMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM<'/)
1002    FORMAT('0',10X,'( Deseas leer las instrucciones  [ S/N ] ?  '\)
1003    FORMAT('0',27X, '( Que opcion eliges ?  '\)
1004    FORMAT (/////'0',15X,
     *  'Actualmente no hay ningun problema en memoria.'/)
1005    FORMAT (
     *  '0',12X,'( Deseas sacar una copia en diskette del modelo ?  '\)
1006    FORMAT (//
     *  '0',17X,     'Paquete Interactivo de Programacion Lineal'////
     *  ' ',15X,   'Derechos reservados por  Enrique Murillo Oth"n'//
     *  ' ',12X,'Augusto Rodin 473, Col. Mixcoac. Mexico 03910, D. F.'//
     * /'0',26X,          '***  Version  9/86  ***'/
     */'0',12X,'Esta es la version para 512 K de PIPL. Tiene capacidad'/
     *' ',10X,'para manejar modelos de hasta 100 restricciones y hasta'/
     *' ',10X,'100 variables de decision (sin contar las de holgura).'/
     */'0',21X,      'Teclea [ RETURN ] para continuar. '\)
        END

        SUBROUTINE MENUS (I)
        IMPLICIT LOGICAL (S)
        COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
        COMMON /FORM/ IMP(8), LUN, FMT
        DIMENSION II(8)
        CHARACTER IMP, II, OP1, OP2
        CHARACTER*8 FMT, MODELO
        CHARACTER*9 UNIA, UNI1, UNI2
        CHARACTER*12 STATUS
        CHARACTER*13 HELP
        CHARACTER*18 ORIGEN
        CALL VITICS (0)
        IF (SINSTR) THEN
                HELP = '
        ELSE
                HELP = 'Instrucciones'
        ENDIF
        IF (I.EQ.1.OR.I.EQ.2.OR.I.EQ.5)
     *          WRITE (*, 1000) MODELO, HELP, STATUS
        IF (I.EQ.1) WRITE (*, 1001)
        IF (I.EQ.2) WRITE (*, 1002)
        IF (I.EQ.3) THEN
                WRITE (*, 1003)
```

```
            WRITE (*, 1031)
      ENDIF
      IF (I.EQ.4) THEN
            DO 10 J = 1, 8
10          II(J) = ' '
            IF (IMP(1).NE.' ') II(1) = '1'
            IF (IMP(2).NE.' ') II(2) = '2'
            IF (IMP(3).NE.' ') II(3) = '3'
            IF (IMP(4).NE.' ') II(4) = '4'
            IF (IMP(5).NE.' ') II(5) = '5'
            IF (IMP(6).NE.' ') II(6) = '6'
            IF (IMP(7).NE.' ') II(7) = '7'
            IF (IMP(8).NE.' ') II(8) = '8'
            IF (LUN.EQ.1) THEN
                  UNIA = 'PANTALLA  '
                  UNI1 = 'Impresora.'
                  UNI2 = 'Diskette. '
                  OP1 = 'I'
                  OP2 = 'D'
            ELSEIF (LUN.EQ.0) THEN
                  UNIA = 'DISKETTE  '
                  UNI1 = 'Pantalla. '
                  UNI2 = 'Impresora.'
                  OP1 = 'P'
                  OP2 = 'I'
            ELSEIF (LUN.EQ.-1) THEN
                  UNIA = 'IMPRESORA '
                  UNI1 = 'Pantalla. '
                  UNI2 = 'Diskette. '
                  OP1 = 'P'
                  OP2 = 'D'
            ENDIF
            IF (FMT.EQ.'ADESEN  ') THEN
                  ORIGEN = ' la Rutina ADESEN.'
            ELSE
                  ORIGEN = 'l Menu Principal. '
            ENDIF
            WRITE (*, 1004) UNIA, (II(J), J = 1, 8),
     *                      OP1, UNI1, OP2, UNI2, ORIGEN
      ENDIF
      IF (I.EQ.5) WRITE (*, 1005)
      IF (I.EQ.6) THEN
            WRITE (*, 1006)
            WRITE (*, 1061) (IMP(J), J = 1, 8)
      ENDIF
      IF (I.EQ.7) WRITE (*, 1061) (IMP(J), J = 1, 8)
      IF (I.EQ.8) WRITE (*, 1007)
      RETURN
1000  FORMAT (' ',2X,'Modelo ',A8,11X,A13,12X,'Solucion : ',A12,/)
1001  FORMAT (/
     */'0',30X,                    'IMMMMMMMMMMMMMMM;'
     */' ',11X,'[1] ACCESO          :  P. I. P. L.  :          [5] ADESEN'
     */' ',13X,  '[2] EDITOR        :               :        [6] GUARDA'
     */' ',15X,    '[3] OPTIMO      :    RUTINAS    :      [7] SALIDA'
     */' ',17X,      '[4] REPORT    HMMMMMMMMMMMMMMM<    [8] AYUDA'
```

```
     */'0',19X,          'Para mayor informacion teclea [RETURN]'/)
 1002    FORMAT (
     * '0',20X,          'IMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM;'
     */' ',20X,          ':                                 :'
     */' ',20X,          ':      \\\\\   \\\   \\\\\   \    :'
     */' ',20X,          ':      [\\\[    [    [\\\[   [    :'
     */' ',20X,          ':      [       \[\   [       [\\\ :'
     */' ',20X,          ':                                 :'
     */' ',20X,          ':                                 :'
     */' ',20X,          ':          MENU PRINCIPAL         :'
     */' ',20X,          ':                                 :'
     */' ',20X,          'HMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM<'
     * //' ',15X,'[1] Lectura de un problema      Rutina ACCESO'
     *  /' ',15X,'[2] Modificar el problema       Rutina EDITOR'
     *  /' ',15X,'[3] Resolver el problema        Rutina OPTIMO'
     *  /' ',15X,'[4] Impresion de reportes       Rutina REPORT'
     *  /' ',15X,'[5] Analisis de Sensibilidad    Rutina ADESEN'
     *  /' ',15X,'[6] Guardar el problema         Rutina GUARDA'
     *  /' ',15X,'[7] Salida del paquete          Rutina SALIDA'
     *  /' ',15X,'[8] Pedir instrucciones         Rutina AYUDA'/)
 1003    FORMAT (
     */'0',20X,           'IMMMMMMMMMMMMMMMMMMMMMMMMMMMMM;'
     */' ',20X,           ':  \\\\\   \\\   \\\\\   \     :'
     */' ',20X,           ':  [\\\[    [    [\\\[   [     :'
     */' ',20X,           ':  [       \[\   [       [\\\\ :'
     */' ',20X,           ':                              :'
     */' ',20X,           ':     EDICION DEL MODELO       :'
     */' ',20X,           'HMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM<')
 1031    FORMAT (
     */' ',9X,'[1]  Listado de los coeficientes de las restricciones.'
     */' ',9X,'[2]  Listado de los coeficientes de la funcion objetivo.'
     */' ',9X,'[3]  Cambios en los coeficientes de las restricciones.'
     */' ',9X,'[4]  Cambios en los Lados Derechos de las restricciones.'
     */' ',9X,'[5]  Cambios en los costos de la funcion objetivo.'
     */' ',9X,'[6]  Cambios en los tipos de restriccion.'
     */' ',9X,'[7]  Cambio de la direccion de la optimizacion.'
     */' ',9X,'[8]  Adicion de una restriccion al modelo.'
     */' ',9X,'[9]  Adicion de una variable al modelo.'
     */' ',9X,'[10] Eliminacion de una restriccion del modelo.'
     */' ',9X,'[11] Eliminacion de una variable del modelo.'
     */' ',9X,'[12] Cambio de nombres o de encabezado del modelo.'
     */'0',14X,   'Teclea [ RETURN ] para regresar al Menu Principal.'/)
 1004    FORMAT (
     */' Unidad de impresion: ',A10,17X,'Resultados a imprimir: ',8A1,//
     */' ',22X,          'IMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM;'
     */' ',22X,          ':                                 :'
     */' ',22X,          ':      \\\\\   \\\   \\\\\   \    :'
     */' ',22X,          ':      [\\\[    [    [\\\[   [    :'
     */' ',22X,          ':      [       \[\   [       [\\\ :'
     */' ',22X,          ':                                 :'
     */' ',22X,          ':                                 :'
     */' ',22X,          ':      IMPRESION DE REPORTES      :'
     */' ',22X,          ':                                 :'
     */' ',22X,          'HMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM<'//
     */' ',15X, '[',A1,'] Imprimir el reporte por ',A10,
```

```
     */' ',15X,  '[',A1,'] Imprimir el reporte por ',A10,
     */' ',15X.  '[R] Seleccionar secciones de resultados a imprimir.'
     */' ',15X,  '[E] Iniciar impresion del reporte.'
     */'0',15X,  'Teclea [ RETURN ] para regresar a',A18,/)
 1005    FORMAT (
     */'0',19X,          'IMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM;'
     */' ',19X,          ':                                   :'
     */' ',19X,          ':       \\\\\   \\\   \\\\\   \      :'
     */' ',19X,          ':       [\\\[    [    [\\\[   [      :'
     */' ',19X,          ':       [       \[\   [       [\\\\  :'
     */' ',19X,          ':                                   :'
     */' ',19X,          ':                                   :'
     */' ',19X,          ':      ANALISIS DE SENSIBILIDAD     :'
     */' ',19X,          ':                                   :'
     */' ',19X,          'HMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM<'
     */'0',13X,'[1]  Cambio en el Lado Derecho de una restriccion.'
     */' ',13X,'[2]  Cambio en la contribucion de alguna variable'
     */' ',18X,     'a la funcion objetivo.'
     */' ',13X,'[3]  Adicion al problema de una nueva variable.'
     */' ',13X,'[4]  Adicion al problema de una nueva restriccion.'
     */'0',13X,'Teclea [ RETURN ] para regresar al Menu Principal.'/)
 1006    FORMAT (
     * '0',19X,          'IMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM;'
     */' ',19X,          ':                                  :'
     */' ',19X,          ':       \\\\\   \\\   \\\\\   \     :'
     */' ',19X,          ':       [\\\[    [    [\\\[   [     :'
     */' ',19X,          ':       [       \[\   [       [\\\  :'
     */' ',19X,          ':                                  :'
     */' ',19X,          ':                                  :'
     */' ',19X,          ':          MENU DE RESULTADOS      :'
     */' ',19X,          ':                                  :'
     */' ',19X,          'HMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM<'/)
 1061    FORMAT (
     * '0',10X,A1,1X,'[1] Valor optimo de la funcion objetivo.'
     */' ',10X,A1,1X,'[2] Valores optimos de las variables de decision.'
     */' ',10X,A1,1X,'[3] Holguras y excesos en las restricciones.'
     */' ',10X,A1,1X,'[4] Valores optimos de las variables duales.'/
     *' ',10X,A1,1X,'[5] Costos reducidos de las variables de decision.'
     */' ',10X,A1,1X,
     *     '[6] Rangos de variacion de costos de la funcion objetivo.'
     */' ',10X,A1,1X,'[7] Rangos de variacion de los Lados Derechos.'
     */' ',10X,A1,1X,'[8] Listado completo del Modelo Lineal.'/)
 1007    FORMAT (
     *//'0',23X,    'IMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM;'
     * /' ',23X,    ':                              :'
     * /' ',23X,    ':    METODO SIMPLEX REVISADO   :'
     * /' ',23X,    ':                              :'
     * /' ',23X,    'HMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM<')
         END

         SUBROUTINE ACCESO
         IMPLICIT LOGICAL (S)
         DOUBLE PRECISION A, B
         COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
         COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
```

```
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      COMMON /FORM/ IMP(8), LUN, FMT
      CHARACTER*8 BNOM, XNOM, ZNOM, FMT, MODELO
      CHARACTER*12 STATUS
      CHARACTER*79 TITULO
      CHARACTER*3 OPT
      CHARACTER IMP, RESP
      CALL VITICS (0)
      IF (.NOT.SINSTR) THEN
              CALL TEXTOS ('ACCESO1.TXT ', 0)
              CALL PAUSA
              CALL ALTO (LIN)
              CALL VITICS (0)
      ENDIF
      IF (M.GT.0) THEN
              IF (.NOT.SINSTR) THEN
                      CALL TEXTOS ('SALIDA.TXT  ', 0)
              ELSE
                      CALL VITICS (4)
              ENDIF
              WRITE (*, 1000)
              READ (*, '(A1)') RESP
              IF (RESP.EQ.'N'.OR.RESP.EQ.'n') RETURN
              CALL INICIA
      ENDIF
5     WRITE (*, 1001)
      READ (*, '(A1)') RESP
      CALL VITICS (0)
      IF (RESP.EQ.'D'.OR.RESP.EQ.'d') THEN
              CALL LEEDISC
              RETURN
      ENDIF
      IF (RESP.NE.'T'.AND.RESP.NE.'t') GO TO 5
      WRITE (*, 1002)
      READ (*, '(A8)') MODELO
      IF (MODELO.EQ.'        ') MODELO = 'de P. L.'
10    WRITE (*, 1004) MODELO
      READ (*, 1003, ERR = 10) M
      IF (M.LT.0) GO TO 10
      IF (M.GT.100) THEN
              CALL TEXTOS ('ERROR1.TXT  ', 1)
              CALL PAUSA
              CALL ALTO (LIN)
              CALL VITICS (0)
              GO TO 10
      ENDIF
      MB = M + 1
      MO = M
15    WRITE (*, 1005)
      READ (*, 1003, ERR = 15) NO
      IF (NO.GT.100) THEN
              CALL TEXTOS ('ERROR1.TXT  ', 1)
              CALL PAUSA
              CALL ALTO (LIN)
```

```
             CALL VITICS (0)
             GO TO 15
        ENDIF
        CALL VITICS (0)
        WRITE (*, 1006)
        READ (*, '(A1)') RESP
        IF (RESP.EQ.'N'.OR.RESP.EQ.'n') GO TO 40
 20     CALL VITICS (0)
        WRITE (*, 1008)
        DO 30 J = 1, NO
        WRITE (*, 1009) J
 30     READ (*, 1010) XNOM(J)
        WRITE (*, 1007)
        READ (*, '(A1)') RESP
        IF (RESP.EQ.'S'.OR.RESP.EQ.'s') GO TO 20
 40     CALL VITICS (0)
        WRITE (*, 1106)
        READ (*, '(A1)') RESP
        IF (RESP.EQ.'N'.OR.RESP.EQ.'n') GO TO 70
 50     CALL VITICS (0)
        WRITE (*, 1108)
        DO 60 I = 1, M
        WRITE (*, 1109) I
 60     READ (*, 1010) BNOM(I)
        WRITE (*, 1007)
        READ (*, '(A1)') RESP
        IF (RESP.EQ.'S'.OR.RESP.EQ.'s') GO TO 50
 70     IF (.NOT.SINSTR) THEN
             CALL TEXTOS ('ACCESO2.TXT ', 0)
        ELSE
             CALL VITICS (0)
             WRITE (*, 1012)
        ENDIF
        DO 80 I = 1, 3
 80     READ (*, '(A79)') TITULO(I)
        CALL VITICS (0)
        WRITE (*, 1015)
        CALL ALTO (LIN)
        DO 90 I = 1, M
        CALL RESTR (I)
        IF (MRESTR(I).NE.1) MB = M + 2
 90     CONTINUE
        CALL VITICS (0)
        WRITE (*, 1013)
        READ (*, 1010) ZNOM
        WRITE (*, 1022) ZNOM
100     WRITE (*, 1023) ZNOM
        DO 120 J = 1, NO
110     WRITE (*, 1016) J, XNOM(J)
120     READ (*, *, ERR = 110) A(MB, J)
        WRITE (*, 1023) ZNOM
        WRITE (*, 1026)
        READ (*, '(A1)') RESP
        IF (RESP.EQ.'S'.OR.RESP.EQ.'s') GO TO 100
130     CALL VITICS (0)
```

```
      WRITE (*, 1024)
      READ (*, '(A3)') OPT
      IF (OPT.EQ.'Min'.OR.OPT.EQ.'MIN'.OR.OPT.EQ.'min') MAXMIN = -1
      IF (OPT.EQ.'Max'.OR.OPT.EQ.'MAX'.OR.OPT.EQ.'max') MAXMIN = 1
      IF (MAXMIN.NE.1.AND.MAXMIN.NE.-1) GO TO 130
      RETURN
1000  FORMAT(/'0',10X,
     * '( Deseas borrar el modelo actual [ S/N ] ?  '\)
1001  FORMAT (////////'0',
     *  . 18X,  '( Que unidad de lectura deseas utilizar ?'
     * //,24X,          '[T] Teclado'
     *  /,24X,          '[D] Disco             Opcion:  '\)
1002  FORMAT (//////'0',10X,
     * '( Cu l es el nombre del modelo (hasta 8 letras) ?  '\)
1003  FORMAT (I3)
1004  FORMAT ('0',10X,
     * '( Cu ntas restricciones tiene el modelo ',A8,' ?  '\)
1005  FORMAT('0',10X,'( Cu ntas variables (sin contar holguras) ?  '\)
1006  FORMAT(///////'0',10X,
     * '( Deseas ponerle nombre a las variables ?  '\)
1007  FORMAT ('0',10X,'( Deseas hacer alguna correccion ?  '\)
1008  FORMAT (//'0',18X,  'Nombres de las Variables de Decisi"n'
     *      /' ',18X,   '------------------------------------'///)
1009  FORMAT (' ',14X,'Variable   X(',I3,' )',6X,'Nombre :  '\)
1106 FORMAT (///////'0',10X,
     *          '( Deseas ponerle nombre a las restricciones ?  '\)
1108  FORMAT (//'0',19X,   'Nombres de las Restricciones'
     *       /' ',19X,    '----------------------------'///)
1109  FORMAT (' ',14X,'Restriccion ',I3,6X,'Nombre :  '\)
1010  FORMAT (A8)
1012  FORMAT (//
    *//'0',24X,         'Encabezado del Problema'
    * /' ',24X,         '-----------------------'
    * /'0',26X,          '(hasta 3 renglones)'
    *//' 0          10         20         30         40         50        6'
    *                                                            ,'0        70'
    * /' |---------|---------|---------|---------|---------|---------|---------|'
    *                                              ,'---------|-------'//)
1013  FORMAT (///'0',10X,
     * '( Que nombre deseas ponerle a la funcion objetivo ?  '\)
1015  FORMAT (/////////
     *  ' ',10X,'  Muy bien. Ahora dame, por favor, los coeficientes'
     * /' ',10X,'de las restricciones.'//
     *  '0',12X,  'Teclea [ RETURN ] para comenzar.  '\)
1016  FORMAT (' ',15X,'X (',I3,' )',5X,A8,5X,'Contribucion : '\)
1022  FORMAT (
    *'0',12X,'Ahora dame los costos o contribuciones de cada'/
    *' ',10X,'variable a la funcion objetivo, ',A8,//)
1023  FORMAT(/' ',20X,'******  Funcion Objetivo  ******'
     *        /' ',32X,A8,/)
1024  FORMAT (////////'0',
     *     16X,'( Como deseas optimizar la funcion objetivo ?'
     *  //' ',22X,    '[MAX] Maximizar'
     *   /' ',22X,    '[MIN] Minimizar              Opcion: '\)
1026  FORMAT ('0',
```

```
     *   12X,'( Deseas hacer alguna correcion [ S/N ] ?  '\)
         END

         SUBROUTINE RESTR (K)
         IMPLICIT LOGICAL (S)
         DOUBLE PRECISION A, B, BI, XB, ZJCJ
         COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
         COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
         COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
         COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
         COMMON /FORM/ IMP(8), LUN, FMT
         CHARACTER IMP, RESP, OPT
         CHARACTER*8 BNOM, XNOM, ZNOM, FMT, MODELO
         CHARACTER*12 STATUS
         CHARACTER*79 TITULO
 10      CALL VITICS (0)
         WRITE (*, 1003) K, BNOM(K)
         DO 30 J = 1, NO
 20      WRITE (*, 1004) J, XNOM(J)
         READ (*, *, ERR = 20) A(K, J)
 30      CONTINUE
         WRITE (*, 1003) K, BNOM(K)
 40      WRITE (*, 1006)
         READ (*, '(I1)', ERR = 40) MRESTR(K)
         IF (MRESTR(K).LT.1.OR.MRESTR(K).GT.3) THEN
                 CALL VITICS (1)
                 GO TO 40
         ENDIF
 50      WRITE (*, 1008)
         READ (*, *, ERR = 50) B(K)
         IF (B(K).LT.0.0D0) GO TO 50
         WRITE (*, 1009)
         READ (*, '(A1)') RESP
         IF (RESP.EQ.'S'.OR.RESP.EQ.'s') GO TO 10
         RETURN
 1002    FORMAT ('    ',A1,/)
 1003    FORMAT (/' ',20X,'******  Restriccion',I4,'  ******'
     *            /,' ',31X,A8,/)
 1004    FORMAT (/'+',13X,'X (',I3,' )',5X,A8,5X,'Coeficiente :  '\)
 1006    FORMAT(/'+',5X,
     *   '[1] s  Menor o igual     [2] r  Mayor o igual     [3] =  Igual'
     *    //' ',12X,'La restriccion es del tipo ( [1], [2] " [3] ?  '\)
 1008    FORMAT(/' ',10X,
     *   'Lado derecho (positivo) de la restriccion :  '\)
 1009    FORMAT ('0',10X,
     *   '( Deseas hacer alguna correccion  [ S/N ] ?  '\)
         END

         SUBROUTINE LEEDISC
         IMPLICIT DOUBLE PRECISION (T), LOGICAL (S)
         DOUBLE PRECISION A, B, BI, XB, ZJCJ
         COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
         COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
         COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
         COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
```

```
      COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
      COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
      COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      CHARACTER*8 BNOM, XNOM, ZNOM, SOL, MODELO
      CHARACTER*12 STATUS
      CHARACTER*79 TITULO
      CHARACTER*14 LPNOM
      CHARACTER*2 FASE
      LOGICAL ARCHIV
      IF (.NOT.SINSTR) THEN
              CALL TEXTOS ('DENTRA.TXT   ', 0)
      ELSE
              CALL VITICS (4)
      ENDIF
      WRITE (*, 999)
10    READ (*, '(A14)') LPNOM
      IF (LPNOM.EQ.'              ') RETURN
      INQUIRE (FILE = LPNOM, EXIST = ARCHIV)
      IF (ARCHIV) THEN
              WRITE (*, 1000)
              OPEN (1, FILE = LPNOM)
      ELSE
              WRITE (*, 998) LPNOM
              GO TO 10
      ENDIF
      READ (1, 1002) MODELO
      READ (1, 1002) SOL
      READ (1, 1003) (TITULO(I), I = 1, 3)
      IF (SOL.EQ.'RESUELTO') THEN
              SOLOPT = .TRUE.
      ELSE
              SOLOPT = .FALSE.
      ENDIF
      READ (1, '(I3)') MO
      M = MO
      READ (1, '(I3)') NO
      IF (MO.GT.100.OR.NO.GT.100) THEN
              CALL TEXTOS ('ERROR1.TXT   ', 1)
              CALL PAUSA
              CALL ALTO (LIN)
              CLOSE (1)
              RETURN
      ENDIF
      READ (1, '(I3)') MAXMIN
      READ (1, 1002) ZNOM
      DO 20 J = 1, NO
20    READ (1, 1002) XNOM(J)
      DO 30 I = 1, MO
30    READ (1, 1001) MRESTR(I), B(I), BNOM(I)
      READ (1, '(I5)') N
      DO 40 K = 1, N
40    READ (1, 1004) I, J, A(I, J)
      MB = I
      READ (1, 1002) SOL
```

```
      IF (SOL.EQ.'SOLUCION') THEN
            READ (1, 1006) T1ZKCK, T2XBYJ, T3YJO
            READ (1, 1006) T4SNF, T5DBI, T6DCJ
            CALL VARHOL
            IF (MB.EQ.MO + 2) M = M + 1
            DO 50 I = 1, MB
50          READ (1, 1005) IBAS(I), XB(I)
            READ (1, '(I5)') N
            DO 60 K = 1, N
60          READ (1, 1004) I, J, BI(I, J)
            CLOSE (1)
            CALL VARNOB
            FASE = 'II'
            METODO = 1
            MFASE = MB
            ITER1 = 0
            ITER2 = 0
            CALL COSTOS
            CALL SRZJCJ (MB)
            CALL RANGOB
            CALL RANGOC
            CALL COSTOS
      ENDIF
      RETURN
998   FORMAT (/'0',12X,'El archivo ',A14,' no existe.'
     * /'0',10X,'Nombre del archivo [ A:NOMBRE.PL ] :  '\)
999   FORMAT (
     *  ' ',12X,'( Cu l es el nombre del archivo donde se encuentra'/
     *  ' ',10X,'guardado el problema a resolver ?'/
     *  '0',10X,'Nombre del archivo [ A:NOMBRE.PL ] :  '\)
1000  FORMAT (//'0',25X,'Leyendo ...  '\)
1001  FORMAT (I3, 5X, D24.17, 5X, A8)
1002  FORMAT (A8)
1003  FORMAT (A79)
1004  FORMAT (I3, 5X, I3, 5X, D24.17)
1005  FORMAT (I3, 5X, D24.17)
1006  FORMAT (3(1X, D24.17))
      END

      SUBROUTINE GUARDA
      IMPLICIT DOUBLE PRECISION (T), LOGICAL (S)
      DOUBLE PRECISION A, B, BI, XB, ZJCJ
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
      COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      CHARACTER*8 BNOM, XNOM, ZNOM, SOL, MODELO
      CHARACTER*12 STATUS
      CHARACTER*79 TITULO
      CHARACTER*14 LPNOM
      LOGICAL ARCHIV
      CALL VITICS (0)
```

```
        IF (.NOT.SINSTR) THEN
                CALL TEXTOS ('GUAR1.TXT    ', 0)
        ELSE
                CALL VITICS (4)
        ENDIF
5       WRITE (*, 999)
        READ (*, '(A14)') LPNOM
        IF (LPNOM.EQ.'              ') RETURN
        INQUIRE (FILE = LPNOM, EXIST = ARCHIV)
        IF (ARCHIV) THEN
                WRITE (*, 1006) LPNOM
                READ (*, '(A1)') RESP
                IF (RESP.NE.'S'.AND.RESP.NE.'s'.AND.RESP.NE.' ') GO TO 5
        ENDIF
        WRITE (*, 1007)
        OPEN (1, FILE = LPNOM, STATUS = 'NEW')
        IF (SOLOPT) THEN
                SOL = 'RESUELTO'
        ELSE
                SOL = '        '
        ENDIF
        WRITE (1, 1002) MODELO
        WRITE (1, 1002) SOL
        WRITE (1, 1003) (TITULO(I), I = 1, 3)
        WRITE (1, '(I3)') MO
        WRITE (1, '(I3)') NO
        WRITE (1, '(I3)') MAXMIN
        WRITE (1, 1002) ZNOM
        DO 10 J = 1, NO
10      WRITE (1, 1002) XNOM(J)
        DO 20 I = 1, MO
20      WRITE (1, 1001) MRESTR(I), B(I), BNOM(I)
        N = 0
        DO 30 I = 1, MB
        DO 30 J = 1, NO
        IF (A(I, J).NE.0.0D0) N = N + 1
30      CONTINUE
        WRITE (1, '(I5)') N
        DO 40 I = 1, MB
        DO 40 J = 1, NO
        IF (A(I, J).NE.0.0D0) WRITE (1, 1004) I, J, A(I, J)
40      CONTINUE
        IF (SOLOPT) THEN
                SOL = 'SOLUCION'
                WRITE (1, 1002) SOL
                WRITE (1, 1000) T1ZKCK, T2XBYJ, T3YJO
                WRITE (1, 1000) T4SNF, T5DBI, T6DCJ
                DO 50 I = 1, MB
50              WRITE (1, 1005) IBAS(I), XB(I)
                N = 0
                DO 60 I = 1, MB
                DO 60 J = 1, MB
                IF (BI(I, J).NE.0.0D0) N = N + 1
60              CONTINUE
                WRITE (1, '(I5)') N
```

```
                DO 70 I = 1, MB
                DO 70 J = 1, MB
                IF (BI(I, J).NE.O.ODO) WRITE (1, 1004) I, J, BI(I, J)
70              CONTINUE
        ELSE
                SOL = '        '
                WRITE (1, 1002) SOL
        ENDIF
        CLOSE (1)
        RETURN
999     FORMAT (
     *    'O',12X,'( En que archivo deseas guardar el problema ?'/
     *    'O',10X,'Nombre del archivo [ A:NOMBRE.PL ] :   '\)
1000    FORMAT (3(1X, D24.17))
1001    FORMAT (I3, 5X, D24.17, 5X, A8)
1002    FORMAT (A8)
1003    FORMAT (A79)
1004    FORMAT (I3, 5X, I3, 5X, D24.17)
1005    FORMAT (I3, 5X, D24.17)
1006    FORMAT ('O',10X,
     *   '( Deseas borrar el antiguo archivo ',A14,'?   '\)
1007    FORMAT (//'O',20X,'Guardando ...    '\)
        END

        SUBROUTINE PAUSA
        DO 10 I = 1, 25000
10      CONTINUE
        WRITE (*, 1000)
        RETURN
1000    FORMAT ('O',22X,'Teclea [ RETURN ] para continuar.'\)
        END

        SUBROUTINE VITICS (I)
        IF (I.GT.O) GO TO 10
        WRITE (*, 1) 27, 91, 50, 74
        RETURN
10      DO 20 J = 1, I
20      WRITE (*, 2)
        RETURN
1       FORMAT (' ', 4A1)
2       FORMAT (' '/)
        END

        SUBROUTINE ALTO (I)
        CHARACTER P
        I = O
        READ (*, '(A1)') P
        RETURN
        END

        SUBROUTINE TEXTOS (TX, I)
        CHARACTER*12 TX
        CHARACTER TEXT(79)
        LIN = O
        CALL VITICS (I)
```

```
      OPEN (1, FILE = TX)
10    READ (1, 1001, END = 20) TEXT
      WRITE (*, 1002) TEXT
      LIN = LIN + 1
      IF (LIN.EQ.25) CALL ALTO (LIN)
      GO TO 10
20    CLOSE (1)
      RETURN
1001  FORMAT(79A1)
1002  FORMAT (' ',79A1)
      END

      SUBROUTINE INICIA
      IMPLICIT LOGICAL (S)
      DOUBLE PRECISION A, B
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      CHARACTER*8 BNOM, XNOM, ZNOM, MODELO
      CHARACTER*12 STATUS
      CHARACTER*79 TITULO
      CALL VITICS (0)
      WRITE (*, 1000)
1000  FORMAT (////////'0',25X,'Inicializando variables ...    '\)
      CALL INIABC (1, 102, 1, 100, 1)
      CALL INIABC (1, 102, 1, 102, 2)
      CALL INIABC (1, 102, 0, 0, 3)
      MODELO = '        '
      SOLOPT = .FALSE.
      SNFACT = .FALSE.
      SNACOT = .FALSE.
      DO 10 I = 1, 100
      BNOM(I) = '        '
10    XNOM(I) = '        '
      M = 0
      MB = 0
      NO = 0
      CALL VITICS (0)
      RETURN
      END

      SUBROUTINE INIABC (II, IF, JI, JF, K)
      DOUBLE PRECISION A, B, BI, XB, ZJCJ
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
      GO TO (100, 200, 300), K
100   DO 110 I = II, IF
      DO 110 J = JI, JF
110   A(I, J) = 0.0D0
      GO TO 400
200   DO 210 I = II, IF
      DO 210 J = JI, JF
210   BI(I, J) = 0.0D0
      GO TO 400
300   DO 310 I = II, IF
```

```
310       B(I) = 0.0D0
400       RETURN
          END

          SUBROUTINE COSTOS
          DOUBLE PRECISION A, B, BI, XB, ZJCJ
          COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
          COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
          COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
          IF (MAXMIN.GT.0) THEN
                  DO 10 J = 1, NO
10                A(MB, J) = -A(MB, J)
          ELSE
                  DO 20 J = 1, NOBAS
20                ZJCJ(J) = -ZJCJ(J)
                  DO 30 I = 1, M
30                BI(MB, I) = -BI(MB, I)
                  XB(MB) = -XB(MB)
          ENDIF
          RETURN
          END

          SUBROUTINE EDITOR
          IMPLICIT LOGICAL (S)
          DOUBLE PRECISION A, B
          COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
          COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
          COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
          CHARACTER*12 STATUS
          CHARACTER*8 MODELO
          CHARACTER RESP
          IF (.NOT.SINSTR) THEN
                  CALL TEXTOS ('EDITMAIN.TXT', 0)
                  CALL PAUSA
                  CALL ALTO (LIN)
          ENDIF
          IF (SOLOPT) THEN
                  CALL TEXTOS ('EDITSOPT.TXT', 0)
                  WRITE (*, 1000)
                  READ (*, '(A1)') RESP
                  IF (RESP.EQ.'S'.OR.RESP.EQ.'s'.OR.RESP.EQ.' ') RETURN
                  M = MO
                  SOLOPT = .FALSE.
                  CALL INIABC (1, MB, 1, MB, 2)
          ENDIF
          IF (SNFACT.OR.SNACOT) THEN
                  CALL INIABC (1, MB, 1, MB, 2)
                  M = MO
                  SNACOT = .FALSE.
                  SNFACT = .FALSE.
          ENDIF
10        CALL MENUS (3)
          WRITE (*, 1001)
          READ (*, '(I2)', ERR = 10) IRESP
          IF (IRESP.EQ.0) RETURN
```

```
      IF (IRESP.EQ.1.OR.IRESP.EQ.2.) CALL EDITL (IRESP)
      IF (IRESP.EQ.3) CALL EDIT3
      IF (IRESP.EQ.4) CALL EDIT4
      IF (IRESP.EQ.5) CALL EDIT5
      IF (IRESP.EQ.6) CALL EDIT6
      IF (IRESP.EQ.7) CALL EDIT7
      IF (IRESP.EQ.8) CALL EDIT8
      IF (IRESP.EQ.9) CALL EDIT9
      IF (IRESP.EQ.10) CALL EDIT10
      IF (IRESP.EQ.11) CALL EDIT11
      IF (IRESP.EQ.12) CALL EDIT12
      GO TO 10
1000  FORMAT('0',12X,'( Deseas volver al Menu Principal [ S/N ] ?  '\)
1001  FORMAT('0',24X,'( Que opcion eliges ?  '\)
      END

      SUBROUTINE EDITL (IRESP)
      DOUBLE PRECISION A, B
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      IF (IRESP.EQ.1) THEN
10            CALL LISTR
20            WRITE (*, 1001)
              READ (*, '(I3)', ERR = 20) I
              IF (I.GT.M.OR.I.LT.0) GO TO 20
              IF (I.EQ.0) RETURN
              CALL VITICS (0)
              CALL LISTV (I)
              GO TO 10
      ELSE
              CALL FUNOBJ
              CALL ALTO (LIN)
              RETURN
      ENDIF
1001  FORMAT(/' ',7X,
     *'Numero de la restriccion a examinar  [ 0 = ninguna ] ?  '
     *\)
      END

      SUBROUTINE LISTV (I1)
      DOUBLE PRECISION A, B
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      COMMON /FORM/ IMP(8), LUN, FMT
      CHARACTER RESP, IMP, PAUSSA
      CHARACTER*2 REL
      CHARACTER*8 BNOM, XNOM, ZNOM, FMT
      CHARACTER*79 TITULO
      IF (LUN.GT.0.OR.FMT.EQ.'        ') THEN
              OPEN (2, FILE = 'CON')
              REL = 's '
              IF (MRESTR(I1).EQ.2) REL = 'r '
              IF (MRESTR(I1).EQ.3) REL = '= '
      ELSE
              OPEN (2, FILE = 'LPT1')
              REL = '<='
```

```
              IF (MRESTR(I1).EQ.2) REL = '>='
              IF (MRESTR(I1).EQ.3) REL = ' ='
         ENDIF
         WRITE (2, 999) I1, BNOM(I1)
         IF (LUN.GT.0) WRITE (2, 1005)
         LIN = 0
         DO 30 J = 1, NO, 5
         IF (LUN.LT.0) THEN
              WRITE (2, 1004)
         ELSE
              WRITE (2, 1005)
         ENDIF
         DO 5 JJ = 0, 4
              K = J + JJ
              IF (K.GT.NO) GO TO 5
              WRITE (2, 1000) XNOM(K)
5        CONTINUE
         IF (LUN.LT.0.OR.K.GT.NO) WRITE (2, 1005)
         DO 10 JJ = 0, 4
              K = J + JJ
              IF (K.GT.NO) GO TO 10
              WRITE (2, 1001) K
10       CONTINUE
         IF (LUN.LT.0.OR.K.GT.NO) THEN
              WRITE (2, 1004)
         ELSE
              WRITE (2, 1005)
         ENDIF
         DO 20 JJ = 0, 4
              K = J + JJ
              IF (K.GT.NO) GO TO 20
              WRITE (2, 1002) A(I1, K)
              JJJ = JJ
20       CONTINUE
         IF (K.LT.NO) WRITE (2, 1005)
         LIN = LIN + 1
         IF (LIN.EQ.3.AND.LUN.GT.0.AND.K + 3.LT.NO) THEN
              READ (*, '(A1)') PAUSSA
              LIN = 0
         ENDIF
30       CONTINUE
         IF (JJJ.EQ.3.OR.JJJ.EQ.4) THEN
              WRITE (2, 1006) REL, B(I1)
         ELSE
              WRITE (2, 1003) REL, B(I1)
         ENDIF
         IF (LUN.GT.0) READ (*, '(A1)') PAUSSA
         CLOSE (2)
         RETURN
999      FORMAT (///'ORESTRICCION :     ',I3,'.-   ',A8,
     *   /' -----------'\)
1000     FORMAT ('     ',A8,'     ',\)
1001     FORMAT ('   X (   ',I2,'  )    ',\)
1002     FORMAT ('   ',1P,G14.7,\)
1003     FORMAT ('   ',1P,A2,' ',G16.9,\)
```

```
1004    FORMAT ('  ',/)
1005    FORMAT ('  ')
1006    FORMAT (' ',///,' ',1P,60X,A2,' ',G16.9,\)
        END

        SUBROUTINE FUNOBJ
        DOUBLE PRECISION A, B
        COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
        COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
        COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
        COMMON /FORM/ IMP(8), LUN, FMT
        CHARACTER RESP, IMP
        CHARACTER*9 OPT
        CHARACTER*8 BNOM, XNOM, ZNOM, FMT
        CHARACTER*79 TITULO
        CALL VITICS (0)
        IF (LUN.GT.0.OR.FMT.EQ.'        ') THEN
                OPEN (2, FILE='CON')
        ELSE
                OPEN (2, FILE='LPT1')
        ENDIF
        IF (MAXMIN.GT.0) THEN
                OPT = 'Maximizar'
        ELSE
                OPT = 'Minimizar'
        ENDIF
        WRITE (2, 999) OPT
        IF (LUN.GT.0) WRITE (2, 1005)
        LIN = 0
        DO 30 J = 1, NO, 5
        IF (LUN.LT.0) THEN
                WRITE (2, 1004)
        ELSE
                WRITE (2, 1005)
        ENDIF
        DO 5 JJ = 0, 4
                K = J + JJ
                IF (K.GT.NO) GO TO 5
                WRITE (2, 1000) XNOM(K)
5       CONTINUE
        IF (LUN.LT.0.OR.K.GT.NO) WRITE (2, 1005)
        DO 10 JJ = 0, 4
                K = J + JJ
                IF (K.GT.NO) GO TO 10
                WRITE (2, 1001) K
10      CONTINUE
        IF (LUN.LT.0.OR.K.GT.NO) THEN
                WRITE (2, 1004)
        ELSE
                WRITE (2, 1005)
        ENDIF
        DO 20 JJ = 0, 4
                K = J + JJ
                IF (K.GT.NO) GO TO 20
                WRITE (2, 1002) A(MB, K)
```

```
20      CONTINUE
        WRITE (2, 1005)
        LIN = LIN + 1
        IF (LIN.EQ.3.AND.LUN.GT.0.AND.K + 5.LT.NO) THEN
                  READ (*, '(A1)') PAUSSA
                  LIN = 0
        ENDIF
30      CONTINUE
        CLOSE (2)
        RETURN
999     FORMAT (/'OFUNCION OBJETIVO :',5X,A9,'  Z  =  ...',
     *  /' ----------------'\)
1000            FORMAT ('     ',A8,'     ',\)
1001            FORMAT ('    X (   ',I2,'  )    ',\)
1002            FORMAT ('   ',1P,G14.7,\)
1004    FORMAT ('  ',/)
1005    FORMAT ('  ')
        END

        SUBROUTINE LISTR
        DOUBLE PRECISION A, B
        COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
        COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
        COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
        COMMON /FORM/ IMP(8), LUN, FMT
        CHARACTER IMP, REL
        CHARACTER*8 BNOM, XNOM, ZNOM, FMT
        CHARACTER*79 TITULO
        CALL VITICS (0)
        WRITE (*, 1000)
        LIN = 0
        DO 10 I = 1, MO
        REL = 's'
        IF (MRESTR(I).EQ.2) REL = 'r'
        IF (MRESTR(I).EQ.3) REL = '='
        WRITE (*, 1001) I, BNOM(I), REL, B(I)
        LIN = LIN + 1
        IF (LIN.EQ.18) CALL ALTO (LIN)
10      CONTINUE
        RETURN
1000    FORMAT (/'0',7X,'Restricciones del Modelo'
     *          /' ',7X,'------------------------'
     *  //' ',10X,'RESTRICCION',8X,'TIPO DE RESTR.',8X,'LADO DERECHO'/)
1001    FORMAT (' ',7X,I3,' .-   ',A8,12X,A1,10X,G18.11)
        END

        SUBROUTINE EDIT3
        DOUBLE PRECISION A, B
        COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
        COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
        COMMON /FORM/ IMP(8), LUN, FMT
        CHARACTER IMP
        CHARACTER*8 BNOM, XNOM, ZNOM, FMT
        CHARACTER*79 TITULO
10      CALL LISTR
```

```
20        WRITE (*, 1000)
          READ (*, '(I3)', ERR = 20) I1
          IF (I1.GT.M.OR.I1.LT.0) GO TO 20
          IF (I1.EQ.0) RETURN
30        CALL VITICS (0)
          CALL LISTV (I1)
40        WRITE (*, 1001)
          READ (*, '(I3)', ERR = 40) J1
          IF (J1.GT.NO.OR.J1.LT.0) GO TO 40
          IF (J1.EQ.0) GO TO 10
50        CALL VITICS (0)
          WRITE (*, 1002) J1, XNOM(J1), I1, BNOM(I1), A(I1, J1)
          READ (*, *, ERR = 50) A(I1, J1)
          GO TO 40
1000      FORMAT (/' ',7X,
     *'( En que restriccion deseas hacer cambios [ 0 = ninguna ] ?  '\)
1001      FORMAT(/' ',7X,
     *    '( Que coeficiente quieres cambiar [ 0 = ninguno ] ?  '\)
1002      FORMAT(/////
     *    //'0',10X,'Coeficiente de la variable  X ( ',I3,' )',3X,A8,/
     *    /' ',10X,'en la restriccion  ',I3,' .-  ',A8,//
     *    /' ',15X,1P,'Valor Actual :    ',G18.11,/
     *    /' ',15X,    'Nuevo Valor  :          '\)
          END

          SUBROUTINE EDIT4
          DOUBLE PRECISION A, B
          COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
          COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
          COMMON /FORM/ IMP(8), LUN, FMT
          CHARACTER RESP, IMP
          CHARACTER*8 BNOM, XNOM, ZNOM, FMT
          CHARACTER*79 TITULO
10        CALL LISTR
20        WRITE (*, 1000)
          READ (*, '(I3)', ERR = 20) L
          IF (L.GT.M.OR.L.LT.0) GO TO 20
          IF (L.EQ.0) RETURN
30        CALL VITICS (0)
          WRITE (*, 1001) L, BNOM(L), B(L)
          READ (*, *, ERR = 30) B(L)
          IF (B(L).LT.0.0) GO TO 30
          GO TO 20
1000      FORMAT (/' ',10X,
     *    'Numero del lado derecho a cambiar [ 0 = ninguno ] : '\)
1001      FORMAT (/////'0',
     *    10X,'Lado derecho de la restriccion:   ',I3,' .-  ',A8,
     *    /'0',25X,1P,  'Valor actual  :  ',G18.11,
     *    /' ',25X,     'Nuevo valor   :          '\)
          END

          SUBROUTINE EDIT5
          DOUBLE PRECISION A, B
          COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
          COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
```

```
      COMMON /FORM/ IMP(8), LUN, FMT
      CHARACTER RESP, IMP
      CHARACTER*8 BNOM, XNOM, ZNOM, FMT
      CHARACTER*79 TITULO
10    CALL FUNOBJ
20    WRITE (*, 1000)
      READ (*, '(I3)', ERR = 20) L
      IF (L.GT.NO.OR.L.LT.0) GO TO 20
      IF (L.EQ.0) RETURN
30    CALL VITICS (0)
      WRITE (*, 1001) L, XNOM(L), A(MB, L)
      READ (*, *, ERR = 30) A(MB, L)
      GO TO 20
1000  FORMAT ('0',10X,
     * 'Numero del costo a cambiar [ 0 = ninguno ] : '\)
1001  FORMAT (/////'0',
     * 10X,'Variable  X (',I3,' )   ',A8,
     * /'0',15X,1P,  'Contribucion actual  :   ',G18.11,
     * /' ',15X,     'Nueva contribucion   :          '\)
      END

      SUBROUTINE EDIT6
      DOUBLE PRECISION A, B
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      CHARACTER RESP
      CHARACTER*8 BNOM, XNOM, ZNOM
      CHARACTER*79 TITULO
10    CALL LISTR
20    WRITE (*, 1000)
      READ (*, '(I3)', ERR = 20) I
      IF (I.LT.0.OR.I.GT.M) GO TO 20
      IF (I.EQ.0) RETURN
30    CALL VITICS (0)
      WRITE (*, 1001) I, BNOM(I), MRESTR(I)
      READ (*, '(I1)', ERR = 30) MRESTR(I)
      IF (MRESTR(I).LT.1.OR.MRESTR(I).GT.3) GO TO 30
      MBN = M + 1
      DO 40 I = 1, M
      IF (MRESTR(I).NE.1) MBN = M + 2
40    CONTINUE
      IF (MBN.NE.MB) THEN
              DO 50 J = 1, NO
              A(MBN, J) = A(MB, J)
50            A(MB, J) = 0.0D0
              MB = MBN
      ENDIF
      GO TO 20
1000  FORMAT ('0',7X,
     * 'Numero de la restriccion a cambiar [ 0 = ninguna ] : '\)
1001  FORMAT (//'0',5X,
     * '[1] s  Menor o igual     [2] r  Mayor o igual     [3] =  Igual'
     *///'0',20X,   '****** Restriccion',I4,'  ******'/,' ',31X,A8,
     * //' ',16X,'La restriccion actualmente es del tipo',I3,
     * //' ',18X,  '( Que tipo de restriccion deseas ? '\)
```

```
      END

      SUBROUTINE EDIT7
      CHARACTER*3 OPT
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      IF (MAXMIN.EQ.1)  OPT = 'MAX'
      IF (MAXMIN.EQ.-1) OPT = 'MIN'
10    CALL VITICS (0)
      WRITE (*, 1000) OPT
      READ (*, '(A3)') OPT
      MAXMIN = 0
      IF (OPT.EQ.'MIN'.OR.OPT.EQ.'min') MAXMIN = -1
      IF (OPT.EQ.'MAX'.OR.OPT.EQ.'max') MAXMIN = 1
      IF (MAXMIN.EQ.0) GO TO 10
      RETURN
1000  FORMAT (///////
     * 'O',20X,  'Sentido actual de la optimizacion :  ',A3,
     * /' ',20X,  'Nuevo  sentido de la optimizacion :  '\)
      END

      SUBROUTINE EDIT8
      DOUBLE PRECISION A, B
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      CHARACTER*8 BNOM, XNOM, ZNOM
      CHARACTER*79 TITULO
10    CALL VITICS (0)
      WRITE (*, 1001)
      READ (*, '(I2)', ERR = 10) N
      IF (N.LE.0) RETURN
      M1 = M + 1
      MN = M + N
      MBN = MB + N
      IF (MN.GT.100) THEN
              CALL TEXTOS ('ERROR1.TXT   ', 1)
              CALL PAUSA
              CALL ALTO (LIN)
              GO TO 10
      ENDIF
      CALL VITICS (0)
      WRITE (*, 1002)
      DO 20 I = M1, MN
      WRITE (*, 1003) I
20    READ (*, '(A8)') BNOM(I)
      CALL VITICS (0)
      WRITE (*, 1004)
      CALL ALTO (LIN)
      DO 30 J = 1, NO
      A(MBN, J) = A(MB, J)
30    A(MB, J) = 0.0D0
      DO 40 I = M1, MN
40    CALL RESTR (I)
      M = MN
      MO = MN
```

```
      MB = MBN
      MBN = M + 1
      DO 50 I = 1, M
      IF (MRESTR(I).NE.1) MBN = M + 2
 50   CONTINUE
      IF (MBN.NE.MB) THEN
              DO 60 J = 1, NO
              A(MBN, J) = A(MB, J)
 60           A(MB, J) = 0.0D0
              MB = MBN
      ENDIF
      RETURN
 1001 FORMAT (/////'0',10X,
     * '( Cu ntas restricciones deseas agregar al modelo ?  '\)
 1002 FORMAT (////'0',10X,
     * '( Cu les son los nombres de las nuevas restricciones ?'//)
 1003 FORMAT(' ',15X,'Restriccion',I4,7X,'Nombre :   '\)
 1004 FORMAT (////////'0',10X,
     * '  Bien.  Dame por favor los coeficientes de las variables'/
     * ' ',10X,'en cada una de las nuevas restricciones.'//
     * ' ',12X,'Teclea [ RETURN ] para empezar.  '\)
      END

      SUBROUTINE EDIT9
      DOUBLE PRECISION A, B
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      COMMON /FORM/ IMP(8), LUN, FMT
      CHARACTER*8 BNOM, XNOM, ZNOM, FMT
      CHARACTER*79 TITULO
      CHARACTER RESP, IMP
 10   CALL VITICS (0)
      WRITE (*, 1001)
      READ (*, '(I2)', ERR = 10) N
      IF (N.LE.0) RETURN
      NO1 = NO + 1
      NON = NO + N
      IF (NON.GT.100) THEN
              CALL TEXTOS ('ERROR2.TXT   ', 1)
              CALL PAUSA
              CALL ALTO (LIN)
              GO TO 10
      ENDIF
      WRITE (*, 1002)
      DO 20 J = NO1, NON
      WRITE (*, 1003) J
 20   READ (*, '(A8)') XNOM(J)
      CALL VITICS (0)
      WRITE (*, 1004) M
      CALL ALTO (LIN)
      DO 50 J = NO1, NON
      CALL VITICS (0)
 30   WRITE (*, 1005) J, XNOM(J)
      DO 40 I = 1, M
 35   WRITE (*, 1006) I, BNOM(I), J
```

```
 40      READ (*, *, ERR = 35) A(I, J)
         WRITE (*, 1005) J, XNOM(J)
 45      WRITE (*, 1007) J, XNOM(J)
         READ (*, *, ERR = 45) A(MB, J)
         WRITE (*, 1008)
         READ (*, '(A1)') RESP
         IF (RESP.EQ.'S'.OR.RESP.EQ.'s') GO TO 30
 50      CONTINUE
         NO = NON
         RETURN
 1001    FORMAT (////' ',10X,
     *   '( Cu ntas variables deseas agregar al modelo ?  '\)
 1002    FORMAT (/'0',10X,
     *   '( Cu les son los nombres de las nuevas variables ?'//)
 1003    FORMAT(' ',15X,'Variable  X(',I3,' )',7X,'Nombre :   '\)
 1004    FORMAT (//////'0',10X,
     *   '   Bien. Dame por favor los coeficientes de las nuevas'/,
     *   ' ',10X,'variables en cada una de las',I4,' restricciones.'//
     *   ' ',12X,'Teclea [ RETURN ] para empezar.   '\)
 1005    FORMAT ('0',21X,'******  Variable',I4,' ******'/' ',32X,A8,/)
 1006    FORMAT (' ',5X,
     *   'Restriccion',I4,2X,A8,4X,'Coeficiente de X(',I3,' ) : '\)
 1007    FORMAT('0',5X,
     *'Contribucion de X(',I3,' ) ',A8,' a la funcion objetivo :  '\)
 1008    FORMAT ('0',10X,'( Deseas hacer alguna correccion [ S/N ] ?   '\)
         END

         SUBROUTINE EDIT10
         DOUBLE PRECISION A, B
         COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
         COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
         COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
         CHARACTER*8 BNOM, XNOM, ZNOM
         CHARACTER*79 TITULO
         CHARACTER RESP
 10      CALL LISTR
 20      WRITE (*, 1000)
         READ (*, '(I3)', ERR = 20) K
         IF (K.GT.M.OR.K.LT.0) GO TO 20
         IF (K.LE.0) RETURN
         MO = MO - 1
         M = M - 1
         DO 40 I = K, M
         DO 30 J = 1, NO
 30      A(I, J) = A(I + 1, J)
         B(I) = B(I + 1)
         MRESTR(I) = MRESTR(I + 1)
         BNOM(I) = BNOM(I + 1)
 40      CONTINUE
         I = M + 1
         CALL INIABC (I, I, 1, NO, 1)
         MRESTR(I) = 0
         BNOM(I) = '
         B(I) = 0.0D0
         MBN = M + 1
```

```
      DO 50 I = 1, M
      IF (MRESTR(I).NE.1) MBN = M + 2
50    CONTINUE
      IF (MBN.NE.MB) THEN
            DO 60 J = 1, NO
            A(MBN, J) = A(MB, J)
60          A(MB, J) = 0.0D0
            MB = MBN
      ENDIF
      GO TO 10
1000  FORMAT ('0',7X,
     * 'Numero de la restriccion a eliminar [ 0 = ninguna ] : '\)
      END

      SUBROUTINE EDIT11
      DOUBLE PRECISION A, B
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      CHARACTER*8 BNOM, XNOM, ZNOM
      CHARACTER*79 TITULO
      CHARACTER RESP
10    CALL FUNOBJ
20    WRITE (*, 1000)
      READ (*, '(I3)', ERR = 20) K
      IF (K.GT.NO.OR.K.LT.0) GO TO 20
      IF (K.EQ.0) RETURN
      NO = NO - 1
      DO 40 J = K, NO
      DO 30 I = 1, M
30    A(I, J) = A(I, J + 1)
      XNOM(J) = XNOM(J + 1)
      A(MB, J) = A(MB, J + 1)
40    CONTINUE
      J = NO + 1
      CALL INIABC (1, M, J, J, 1)
      XNOM(J) = '        '
      A(MB, J) = 0.0D0
      GO TO 10
1000  FORMAT ('0',7X,
     * 'Numero de la variable a eliminar [ 0 = ninguna ] : '\)
      END

      SUBROUTINE EDIT12
      IMPLICIT LOGICAL (S)
      DOUBLE PRECISION A, B
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      CHARACTER RESP
      CHARACTER*8 BNOM, XNOM, ZNOM, MODELO
      CHARACTER*12 STATUS
      CHARACTER*79 TITULO
10    CALL TEXTOS ('EDIT12.TXT  ', 0)
20    WRITE (*, 1001)
      READ (*, '(A1)') RESP
```

```
      CALL VITICS (0)
      IF (RESP.EQ.' ') RETURN
      IF (RESP.EQ.'1') THEN
              WRITE (*, 1002) ZNOM
              READ (*, '(A8)') ZNOM
      ENDIF
      IF (RESP.EQ.'2') THEN
200           CALL LISTR
210           WRITE (*, 1003)
              READ (*, '(I3)', ERR = 210) I
              IF (I.LT.0.OR.I.GT.M) GO TO 210
              IF (I.EQ.0) GO TO 10
              CALL VITICS (0)
              WRITE (*, 1004) I, BNOM(I)
              READ (*, '(A8)') BNOM(I)
              GO TO 210
      ENDIF
      IF (RESP.EQ.'3') THEN
300           CALL FUNOBJ
310           WRITE (*, 1003)
              READ (*, '(I3)', ERR = 310) I
              IF (I.LT.0.OR.I.GT.NO) GO TO 310
              IF (I.EQ.0) GO TO 10
              CALL VITICS (0)
              WRITE (*, 1005) I, XNOM(I)
              READ (*, '(A8)') XNOM(I)
              GO TO 310
      ENDIF
      IF (RESP.EQ.'4') THEN
              WRITE (*, 1006) MODELO
              READ (*, '(A8)') MODELO
              IF (MODELO.EQ.'        ') MODELO = 'de P. L.'
      ENDIF
      IF (RESP.EQ.'5') THEN
400           WRITE (*, 1007) (TITULO(I), I = 1, 3)
              DO 410 I = 1, 3
410           READ (*, '(A79)') TITULO(I)
      ENDIF
      GO TO 10
1001  FORMAT('0',24X, '( Que opcion eliges ?  '\)
1002  FORMAT (////'0',10X,'******  Funcion Objetivo  *****'
     *     ///' ',15X,'Nombre actual :  ',A8,
     *       /' ',15X,'Nuevo  nombre :  '\)
1003  FORMAT ('0',7X,
     * 'Teclea el numero del nombre a cambiar [ 0 = ninguno ] :  '\)
1004  FORMAT (///////'0',20X,'******  Restriccion',I4,'  *****'
     *     ///' ',25X,'Nombre actual :  ',A8,
     *       /' ',25X,'Nuevo  nombre :  '\)
1005  FORMAT (///////'0',20X,'******  Variable',I4,'  *****'
     *     ///' ',25X,'Nombre actual :  ',A8,
     *       /' ',25X,'Nuevo  nombre :  '\)
1006  FORMAT (///////'0',20X,'******  Modelo de P. L.  *****'
     *     ///' ',25X,'Nombre actual :  Modelo ',A8,
     *       /' ',25X,'Nuevo  nombre :  Modelo '\)
1007  FORMAT (/'0',28X,'El encabezado actual es :'//
```

```
     *          '0',A79,/,'0',A79,/,'0',A79,///,
     *      ' ',19X,'Escribe a continuacion el nuevo encabezado :'
     *//'0',24X,        'Encabezado del Problema'
     * /' ',24X,        '-----------------------'
     *//' 0         10        20        30        40        50        6'
     *                                                   ,'0        70'
     * /' |---------|---------|---------|---------|---------|---------|'
     *                                ,'---------|-------'//)
      END

      SUBROUTINE OPTIMO
      IMPLICIT DOUBLE PRECISION (T, X, Y, Z), LOGICAL (S)
      DOUBLE PRECISION A, B, BI
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
      COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
      COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJ0, T4SNF, T5DBI, T6DCJ
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      CHARACTER*8 BNOM, XNOM, ZNOM, MODELO
      CHARACTER*12 STATUS
      CHARACTER*79 TITULO
      CHARACTER RESP
      CHARACTER*2 FASE
      CALL VITICS (0)
      CALL PRECIS
      CALL VARHOL
      IF (.NOT.SINSTR) THEN
              CALL TEXTOS ('OPTIM1.TXT  ', 0)
              CALL PAUSA
              CALL ALTO (LIN)
      ENDIF
      NFREC = 0
      IF (M.GT.25) THEN
              IF (.NOT.SINSTR) THEN
                      CALL TEXTOS ('MAXPREC1.TXT', 0)
              ELSE
                      CALL VITICS (5)
              ENDIF
              WRITE (*, 1003)
              READ (*, '(A1)') RESP
              IF (RESP.EQ.'S'.OR.RESP.EQ.'s') THEN
                   IF (.NOT.SINSTR) CALL TEXTOS ('MAXPREC2.TXT', 0)
5                  WRITE (*, 1004)
                   READ (*, '(I3)', ERR = 5) NFREC
              ENDIF
      ENDIF
      CALL VARBAS (FASE)
      CALL MENUS (B)
      METODO = 1
      MFASE = M + 1
      ITER1 = 0
10    CALL SIMREV (NFREC, FASE, MFASE, K, ITER)
```

```
      IF (SNACOT) THEN
              CALL TEXTOS ('OPTIM2.TXT   ', 0)
              WRITE (*, 1001) K, XNOM(K)
              CALL PAUSA
              CALL ALTO (LIN)
              RETURN
      ENDIF
      IF (SNFACT) THEN
              CALL TEXTOS ('OPTIM3.TXT   ', 0)
              CALL PAUSA
              CALL ALTO (LIN)
              RETURN
      ENDIF
      IF (FASE.EQ.'I ') THEN
              ITER1 = ITER
              M = M + 1
              MFASE = MB
              FASE = 'II'
              GO TO 10
      ENDIF
      ITER2 = ITER
      SOLOPT = .TRUE.
      WRITE (*, 1002)
      CALL RANGOB
      CALL RANGOC
      CALL ALTO (LIN)
      RETURN
1001  FORMAT ('0',20X,'La variable no acotada es  X(',I2,')     ',A8,//)
1002  FORMAT (//'0',30X,      'Optimo alcanzado.'///
     *        ' ',22X,'Teclea [ RETURN ] para continuar.')
1003  FORMAT ('0',10X,'( Deseas trabajar con m xima precisi"n ?   '\)
1004  FORMAT('0',10X,
     *  '   ( Despues de cu ntas iteraciones deseas que se recalcule'
     *  ' ',10X,'la matriz inversa ?   '\)
      END

      SUBROUTINE VARHOL
      DOUBLE PRECISION A, B
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      CHARACTER*8 BNOM, XNOM, ZNOM
      CHARACTER*79 TITULO
      NHPOS = 0
      NHNEG = 0
      NRE = 0
      DO 5 I = 1, M
              IF (MRESTR(I).EQ.1) NHPOS = NHPOS + 1
              IF (MRESTR(I).EQ.2) NHNEG = NHNEG + 1
              IF (MRESTR(I).EQ.3) NRE = NRE + 1
5     CONTINUE
      NH = NHPOS + NHNEG
      NOH = NO + NH
```

```
      NA = NHNEG + NRE
      NOHA = NOH + NA
      K = 0
      L = 0
      DO 40 I = 1, M
      GO TO (10, 20, 30), MRESTR(I)
10    K = K + 1
      IHOL(K, 1) = I
      IHOL(K, 2) = 1
      GO TO 40
20    K = K + 1
      L = L + 1
      IHOL(K, 1) = I
      IHOL(K, 2) = -1
      IHOL(NH + L, 1) = I
      IHOL(NH + L, 2) = 1
      GO TO 40
30    L = L + 1
      IHOL(NH + L, 1) = I
      IHOL(NH + L, 2) = 1
40    CONTINUE
      IHOL(NH + NA + 1, 1) = M + 1
      IHOL(NH + NA + 1, 2) = 1
      IF (NA.GT.0) THEN
              BNOM(M + 1) = 'VARS-ART'
              MRESTR(M + 1) = 3
              B(M + 1) = 0.0D0
              IHOL(NH + NA + 2, 1) = M + 2
              IHOL(NH + NA + 2, 2) = 1
      ENDIF
      RETURN
      END

      SUBROUTINE VARBAS (FASE)
      IMPLICIT DOUBLE PRECISION (T, X, Y, Z), LOGICAL (S)
      DOUBLE PRECISION A, B, BI
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
      COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      CHARACTER*12 STATUS
      CHARACTER*8 MODELO
      CHARACTER*2 FASE
      K = 0
      L = 0
      XB(M + 1) = 0.0D0
      DO 40 I = 1, M
      GO TO (10, 20, 30), MRESTR(I)
10    K = K + 1
      IBAS(I) = NO + K
      BI(I, I) = 1.0D0
      XB(I) = B(I)
      GO TO 40
```

```
20      K = K + 1
        L = L + 1
        IBAS(I) = NOH + L
        BI(I, I) = 1.0D0
        BI(M + 1, I) = -1.0D0
        XB(I) = B(I)
        XB(M + 1) = XB(M + 1) - XB(I)
        GO TO 40
30      L = L + 1
        IBAS(I) = NOH + L
        BI(I, I) = 1.0D0
        BI(M + 1, I) = -1.0D0
        XB(I) = B(I)
        XB(M + 1) = XB(M + 1) - XB(I)
40      CONTINUE
        B(M + 1) = 0.0D0
        MRESTR(M + 1) = 3
        BI(M + 1, M + 1) = 1.0D0
        IBAS(M + 1) = NOHA + 1
        FASE = 'II'
        IF (NA.GT.0) THEN
                BI(MB, MB) = 1.0D0
                IBAS(MB) = NOHA + 2
                XB(MB) = 0.0D0
                FASE = 'I '
        ENDIF
        CALL VARNOB
        RETURN
        END

        SUBROUTINE VARNOB
        COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
        COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
        MI = MO + 1
        L = 0
        DO 30 J = 1, 200
10      L = L + 1
        DO 20 I = 1, MI
        IF (L.EQ.IBAS(I)) GO TO 10
20      CONTINUE
        IF (L.LE.NOH) THEN
                INBAS(J) = L
        ELSE
                NOBAS = J - 1
                RETURN
        ENDIF
30      CONTINUE
        RETURN
        END

        SUBROUTINE BASE
        IMPLICIT DOUBLE PRECISION (T, X, Z), LOGICAL (S)
        DOUBLE PRECISION A, AUX, B, BI
        COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
        COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
```

```
      COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
      COMMON /REINVR/ AUX(102, 102)
      CHARACTER*12 STATUS
      CHARACTER*8 MODELO
      DO 30 J = 1, MB
      IF (IBAS(J).GT.NO) THEN
              L = IBAS(J) - NO
              DO 10 I = 1, MB
10            AUX(I, J) = 0.0D0
              IF (IHOL(L, 2).GT.0) THEN
                      AUX(IHOL(L, 1), J) = 1.0D0
              ELSE
                      AUX(IHOL(L, 1), J) = -1.0D0
              ENDIF
      ELSE
              DO 20 I = 1, MB
20            AUX(I, J) = A(I, IBAS(J))
      ENDIF
30    CONTINUE
      CALL INIABC (1, MB, 1, MB, 2)
      DO 40 I = 1, MB
40    BI(I, I) = 1.0D0
      CALL SLINEQ (MB)
      DO 50 I = 1, MB
      XB(I) = 0.0D0
      DO 50 J = 1, MB
50    XB(I) = XB(I) + BI(I, J) * B(J)
      RETURN
      END

      SUBROUTINE SLINEQ (MB)
      DOUBLE PRECISION AUX, BI, XB, ZJCJ, U(102, 3), UMAX, V
      COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
      COMMON /REINVR/ AUX(102, 102)
      JMAX = 0
      NEQ = MB
      NRHS = MB
C
C     Escalamiento de las ecuaciones
C
      DO 80 I = 1, NEQ
      UMAX = 0.0D0
      DO 90 J = 1, NEQ
      V = DABS(AUX(I, J))
90    IF (V.GT.UMAX) UMAX = V
      UMAX = 8.0D0 * UMAX
      DO 110 J = 1, NEQ
110   AUX(I, J) = AUX(I, J) / UMAX
      DO 80 J = 1, NRHS
80    BI(I, J) = BI(I, J) / UMAX
C
```

```
C        Triangulacion de la matriz A
C
70       DO 150 I = 1, NEQ
C
C        Calculo de numeradores de elementos de la matriz L
C
         I1 = I + 1
         I2 = I - 1
         UMAX = 0.0D0
         DO 160 J = I, NEQ
         U(J, 1) = AUX(J, I)
         IF (I.EQ.1) GO TO 170
         DO 180 K = 1, I2
         V = AUX(K, I)
180      U(J, 1) = U(J, 1) - V * AUX(J, K)
170      V = DABS(U(J, 1))
         IF (V.LE.UMAX) GO TO 160
         UMAX = V
         JMAX = J
160      CONTINUE
         Prueba = 1 / Umax
         IF (JMAX.EQ.I) GO TO 190
C
C        Intercambio de renglones
C
         V = U(JMAX, 1)
         U(JMAX, 1) = U(I, 1)
         U(I, 1) = V
         DO 200 J = 1, NEQ
         V = AUX(JMAX, J)
         AUX(JMAX, J) = AUX(I, J)
200      AUX(I, J) = V
         DO 210 J = 1, NRHS
         V = BI(JMAX, J)
         BI(JMAX, J) = BI(I, J)
210      BI(I, J) = V
C
C        Calculo de los elementos de la diagonal
C
190      AUX(I, I) = DSQRT(DABS(U(I, 1)))
         U(I, 1) = U(I, 1) / AUX(I, I)
C
C        Calculo de los elementos de la matriz L
C
         IF (I.EQ.NEQ) GO TO 150
         DO 220 J = I1, NEQ
         AUX(J, I) = U(J, 1) / U(I, 1)
C
C        Calculo de los elementos de la matriz U
C
         UMAX = AUX(I, J)
         IF (I.EQ.1) GO TO 220
         DO 230 K = 1, I2
         V = AUX(K, J)
230      UMAX = UMAX - V * AUX(I, K)
```

```
220     AUX(I, J) = UMAX / AUX(I, I)
150     CONTINUE
C
C       Procedimiento iterativo sobre los residuos
C
        DO 250 IRHS = 1, NRHS
        DO 260 I = 1, NEQ
260     U(I, 3) = 0.0D0
        DO 270 ITNO = 1, 40
        ICONV = 0
        DO 280 I2 = 1, NEQ
        I = NEQ - I2 + 1
        U(I, 2) = BI(I, IRHS)
        DO 280 K = 1, I
        UMAX = U(K, 1) * U(K, 3)
        IF (K.EQ.NEQ) GO TO 280
        I1 = K + 1
        DO 290 J = I1, NEQ
290     UMAX = UMAX + AUX(K, J) * U(J, 3)
280     U(I, 2) = U(I, 2) - AUX(I, K) * UMAX
C
C       Solucion de las ecuaciones
C
        DO 300 I = 1, NEQ
        I2 = I - 1
        IF (I.EQ.1) GO TO 300
        DO 310 K = 1, I2
310     U(I, 2) = U(I, 2) - AUX(I, K) * U(K, 2)
300     U(I, 2) = U(I, 2) / AUX(I, I)
        DO 320 I2 = 1, NEQ
        I = NEQ - I2 + 1
        I1 = I + 1
        IF (I.EQ.NEQ) GO TO 330
        DO 340 K = I1, NEQ
340     U(I, 2) = U(I, 2) - AUX(I, K) * U(K, 2)
330     U(I, 2) = U(I, 2) / U(I, 1)
        IF (DABS(U(I, 2)).GT.1.0D-10 * DABS(U(I, 3))) ICONV = 1
320     U(I, 3) = U(I, 3) + U(I, 2)
        IF (ICONV.EQ.0) GO TO 360
270     CONTINUE
360     DO 250 I = 1, NEQ
250     BI(I, IRHS) = U(I, 3)
        RETURN
        END

        SUBROUTINE PRECIS
        IMPLICIT DOUBLE PRECISION (T)
        DOUBLE PRECISION A, B
        COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
        COMMON /VECTDP/ T1(102), T2(102)
        COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
        T2XBYJ= -1.0D5
        T3YJO = 1.0D0
        T4SNF = 1.0D0
        T5DBI = -1.0D5
```

```
      T6DCJ = -1.0D5
      DO 10 I = 1, M
10    T1(I) = -B(I)
      CALL MINIMO (T1, T5DBI, M, L)
C     T5+ Parametro de cambio en bi ==> rangob(i) no acotado
      T5DBI = -T5DBI * 1.0D5
      DO 20 J = 1, NO
20    T1(J) = -DABS(A(MB, J))
      CALL MINIMO (T1, T6DCJ, NO, L)
C     T6+ Parametro de cambio en ci ==> rangoc(i) no acotado
      T6DCJ = -T6DCJ * 1.0D5
C     T1- Parametro de minimo precio sombra que causa otra iteracion
      T1ZKCK = -1.0D2 / T6DCJ
      DO 40 J = 1, NO
      DO 30 I = 1, M
      IF (DABS(A(I, J)).LT.1.0D-08) THEN
              T1(I) = 0.0D0
      ELSE
              T1(I) = -B(I) / DABS(A(I, J))
      ENDIF
      T2(I) = 1.0D0
      IF (B(I).EQ.0.0D0.OR.A(I, J).EQ.0.0D0) GO TO 30
      T2(I) = B(I) / DABS(A(I, J))
30    CONTINUE
      CALL MINIMO (T1, T2XBYJ, M, L)
      CALL MINIMO (T2, T4SNF, M, L)
40    CONTINUE
C     T2+ Parametro de maximo valor de var. bas. ==> no acotada
      T2XBYJ = -T2XBYJ * 1.0D5
C     T4- Parametro de minimo valor negativo permitido para var. bas.
      T4SNF  = -T4SNF / 1.0D4
      DO 60 J = 1, NO
      DO 50 I = 1, M
              IF (A(I, J).EQ.0.0D0) THEN
                      T1(I) = 2.0D0
              ELSE
                      T1(I) = DABS(A(I, J))
              ENDIF
50    CONTINUE
      CALL MINIMO (T1, T3YJO, M, L)
60    CONTINUE
C     T3+ Parametro de minimo valor permitido elemento BI ==> = 0.
      T3YJO = 1.0D-07 * T3YJO
      RETURN
      END

      SUBROUTINE SIMREV (NFREC, FASE, MFASE, K, ITER)
      IMPLICIT DOUBLE PRECISION (T, X, Y, Z), LOGICAL (S)
      DOUBLE PRECISION A, B, BI
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
      COMMON /VECTDP/ T1(102), YJ(102)
      COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
```

```
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      CHARACTER*12 STATUS
      CHARACTER*8 MODELO
      CHARACTER*2 FASE
      ITER = 0
      NINV = 0
      ITMAX = 2 * (MB + NO)
      SNACOT = .FALSE.
      SNFACT = .FALSE.
      WRITE (*, 1001) FASE
 20   IF (FASE.EQ.'I '.AND.XB(MFASE).GT.T4SNF) RETURN
      CALL SRZJCJ (MFASE)
      ZKCK = T1ZKCK
      CALL MINIMO (ZJCJ, ZKCK, NOBAS, IE)
      IF (IE.EQ.0) THEN
              IF (FASE.EQ.'I '.AND.XB(MFASE).LT.T4SNF) SNFACT = .TRUE.
              RETURN
      ENDIF
      ITER = ITER + 1
      NINV = NINV + 1
      WRITE (*, 1003) ITER
      K = INBAS(IE)
      DO 40 I = 1, MB
 40   YJ(I) = 0.0D0
      IF (K.LE.NO) THEN
              DO 60 I = 1, M
              DO 50 J = 1, MB
                      IF (DABS(BI(I, J)).LT.T3YJO) GO TO 50
                      YJ(I) = YJ(I) + BI(I, J) * A(J, K)
 50           CONTINUE
              IF (YJ(I).LE.T3YJO) THEN
                      T1(I) = T2XBYJ * 2.0D0
              ELSE
                      T1(I) = XB(I) / YJ(I)
              ENDIF
 60           CONTINUE
              IF (NA.GT.0) THEN
                      DO 70 I = 1, MB
                        IF (DABS(BI(MB, I)).LT.T3YJO) GO TO 70
                        YJ(MB) = YJ(MB) + BI(MB, I) * A(I, K)
 70                   CONTINUE
              ENDIF
     ELSE
              INB = K - NO
              IF (IHOL(INB, 2).GT.0) THEN
                      DO 80 I = 1, M
 80                   YJ(I) = BI(I, IHOL(INB, 1))
              ELSE
                      DO 90 I = 1, M
 90                   YJ(I) = -BI(I, IHOL(INB, 1))
              ENDIF
              DO 100 I = 1, M
              IF (YJ(I).LE.T3YJO) THEN
                      T1(I) = T2XBYJ * 2.0D0
              ELSE
```

```
                         T1(I) = XB(I) / YJ(I)
                 ENDIF
  100            CONTINUE
                 IF (NA.GT.0) THEN
                         IF (IHOL(INB, 2).GT.0) THEN
                                 YJ(MB) = BI(MB, IHOL(INB, 1))
                         ELSE
                                 YJ(MB) = -BI(MB, IHOL(INB, 1))
                         ENDIF
                 ENDIF
         ENDIF
         YJ(MFASE) = ZKCK
         T = T2XBYJ
         CALL MINIMO (T1, T, M, IR)
         IF (IR.EQ.0) THEN
                 SNACOT = .TRUE.
                 RETURN
         ENDIF
         IF (IBAS(IR).GT.NOH) THEN
                 INBAS(IE) = INBAS(NOBAS)
                 INBAS(NOBAS) = IBAS(IR)
                 IBAS(IR) = K
                 NOBAS = NOBAS - 1
         ELSE
                 INBAS(IE) = IBAS(IR)
                 IBAS(IR) = K
         ENDIF
         IF (MFASE.EQ.MB.AND.NINV.EQ.NFREC) THEN
                 CALL BASE
                 NINV = 0
                 GO TO 20
         ENDIF
         CALL ITERA (IR)
         IF (MFASE.EQ.MB.AND.ITER.GT.ITMAX) THEN
                 CALL TEXTOS ('OPTIM4.TXT  ', 2)
                 CALL ALTO (LIN)
                 RETURN
         ENDIF
         GO TO 20
 1001    FORMAT (//' ',24X,'   0 Iteraciones de Fase ',A2)
 1003    FORMAT ('+',25X,I3)
         END

         SUBROUTINE SRZJCJ (MFASE)
         IMPLICIT DOUBLE PRECISION (T, X, Y, Z)
         DOUBLE PRECISION A, B, BI
         COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
         COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
         COMMON /VECTDP/ T1(102), YJ(102)
         COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
         COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
         COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
         DO 30 J = 1, NOBAS
         IF (INBAS(J).GT.NO) THEN
                 INB = INBAS(J) - NO
```

```
              IF (IHOL(INB, 2).GT.0) THEN
                      ZJCJ(J) = BI(MFASE, IHOL(INB, 1))
              ELSE
                      ZJCJ(J) = -BI(MFASE, IHOL(INB, 1))
              ENDIF
      ELSE
              ZJCJ(J) = 0.0D0
              DO 20 I = 1, MB
20            ZJCJ(J) = ZJCJ(J) + BI(MFASE, I) * A(I, INBAS(J))
      ENDIF
30    CONTINUE
      RETURN
      END

      SUBROUTINE ITERA (IR)
      IMPLICIT DOUBLE PRECISION (T, X, Y, Z)
      DOUBLE PRECISION A, B, BI
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, ND
      COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
      COMMON /VECTDP/ T1(102), YJ(102)
      COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
      DO 10 J = 1, MB
10    BI(IR, J) = BI(IR, J) / YJ(IR)
      XB(IR) = XB(IR) / YJ(IR)
      DO 30 I = 1, MB
              IF (I.EQ.IR.OR.DABS(YJ(I)).LT.T3YJO) GO TO 30
              DO 20 J = 1, MB
20            BI(I, J) = BI(I, J) - YJ(I) * BI(IR, J)
              XB(I) = XB(I) - YJ(I) * XB(IR)
30    CONTINUE
      RETURN
      END

      SUBROUTINE DUALSI
      IMPLICIT DOUBLE PRECISION (T, X, Y, Z), LOGICAL (S)
      DOUBLE PRECISION A, B, BI
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, ND
      COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
      COMMON /VECTDP/ T1(102), YJ(102)
      COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
      COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      CHARACTER*12 STATUS
      CHARACTER*8 MODELO
      ITER1 = 0
      CALL VITICS (0)
      WRITE (*, 1001)
10    XMIN = T4SNF
      CALL MINIMO (XB, XMIN, M, IR)
      IF (IR.EQ.0) THEN
              WRITE (*, 1002)
              RETURN
      ENDIF
```

```
      DO 30 J = 1, NOBAS
      IF (INBAS(J).LE.NO) THEN
              YJ(J) = 0.0D0
              DO 20 I = 1, M
20            YJ(J) = YJ(J) + BI(IR, I) * A(I, INBAS(J))
      ELSE
              INB = INBAS(J) - NO
              IF (IHOL(INB, 2).GT.0) THEN
                      YJ(J) = BI(IR, IHOL(INB, 1))
              ELSE
                      YJ(J) = -BI(IR, IHOL(INB, 1))
              ENDIF
      ENDIF
      IF (YJ(J).GT.-T3YJ0) THEN
              T1(J) = 2.0D0 * T2XBYJ
      ELSE
              T1(J) = -ZJCJ(J) / YJ(J)
      ENDIF
30    CONTINUE
      T = T2XBYJ
      CALL MINIMO (T1, T, NOBAS, IE)
      IF (IE.EQ.0) THEN
              SNFACT = .TRUE.
              RETURN
      ENDIF
      ITER1 = ITER1 + 1
      K = INBAS(IE)
      IF (K.LE.NO) THEN
              DO 40 I = 1, MB
              YJ(I) = 0.0D0
              DO 40 J = 1, MB
40            YJ(I) = YJ(I) + BI(I, J) * A(J, K)
      ELSE
              INB = K - NO
              IF (IHOL(INB, 2).GT.0) THEN
                      DO 50 I = 1, MB
50                    YJ(I) = BI(I, IHOL(INB, 1))
              ELSE
                      DO 60 I = 1, MB
60                    YJ(I) = -BI(I, IHOL(INB, 1))
              ENDIF
      ENDIF
      IF (IBAS(IR).GT.NOH) THEN
              INBAS(IE) = INBAS(NOBAS)
              INBAS(NOBAS) = IBAS(IR)
              IBAS(IR) = K
              NOBAS = NOBAS - 1
      ELSE
              INBAS(IE) = IBAS(IR)
              IBAS(IR) = K
      ENDIF
      CALL ITERA (IR)
      CALL SRZJCJ (MB)
      WRITE (*, 1003) ITER1
      GO TO 10
```

```
1001      FORMAT (//
     *//'0',25X,   'IMMMMMMMMMMMMMMMMMMMMMMMMMM;'
     * /' ',25X,   ':                          :'
     * /' ',25X,   ':     METODO DUAL SIMPLEX   :'
     * /' ',25X,   ':                          :'
     * /' ',25X,   'HMMMMMMMMMMMMMMMMMMMMMMMMMM<'
     *//'0',30X,          '   O Iteraciones')
1002      FORMAT (//'0',30X,'Optimo alcanzado.'///
     *            ' ',22X,'Teclea [ RETURN ] para continuar.   '\)
1003      FORMAT ('+',30X,I3)
          END

          SUBROUTINE RANGOB
          IMPLICIT DOUBLE PRECISION (R, T, X, Y, Z)
          DOUBLE PRECISION A, B, BI
          COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
          COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
          COMMON /VECTDP/ T1(102), T2(102)
          COMMON /RANGOS/ RBSUP(100), RBINF(100), RCSUP(102), RCINF(102)
          COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
          COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
          COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
          COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
          DO 50 K = 1, MO
          RBN = 2.0D0 * T5DBI
                  DO 40 I = 1, M
                  IF (BI(I, K).LE.-T3YJO) THEN
                          T1(I) = -XB(I) / BI(I, K)
                          T2(I) = RBN
                  ELSEIF (DABS(BI(I, K)).LT.T3YJO) THEN
                          T1(I) = RBN
                          T2(I) = RBN
                  ELSEIF (BI(I, K).GE.T3YJO) THEN
                          T2(I) = XB(I) / BI(I, K)
                          T1(I) = RBN
                  ENDIF
40                CONTINUE
          RBN = T5DBI
          CALL MINIMO (T1, RBN, M, L)
          RBSUP(K) = B(K) + RBN
          RBN = T5DBI
          CALL MINIMO (T2, RBN, M, L)
          RBINF(K) = B(K) - RBN
50        CONTINUE
          RETURN
          END

          SUBROUTINE RANGOC
          IMPLICIT DOUBLE PRECISION (R, T, X, Y, Z)
          DOUBLE PRECISION A, B, BI
          COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
          COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
          COMMON /VECTDP/ T1(102), T2(102)
          COMMON /RANGOS/ RBSUP(100), RBINF(100), RCSUP(102), RCINF(102)
          COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
```

```
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
      DO 30 K = 1, M
      IF (IBAS(K).GT.NO) GO TO 30
      DO 20 J = 1, NOBAS
      IF (INBAS(J).GT.NO) THEN
              INB = INBAS(J) - NO
              IF (IHOL(INB, 2).GT.0) THEN
                      YJ = BI(K, IHOL(INB, 1))
              ELSE
                      YJ = -BI(K, IHOL(INB, 1))
              ENDIF
      ELSE
              YJ = 0.0D0
              DO 10 I = 1, M
10            YJ = YJ + BI(K, I) * A(I, INBAS(J))
      ENDIF
      T1(J) = 2.0D0 * T6DCJ
      IF (YJ.LT.-T3YJO) T1(J) = -ZJCJ(J) / YJ
      T2(J) = 2.0D0 * T6DCJ
      IF (YJ.GT.T3YJO) T2(J) = ZJCJ(J) / YJ
20    CONTINUE
      RC1 = T6DCJ
      RC2 = T6DCJ
      CALL MINIMO (T1, RC1, NOBAS, L)
      CALL MINIMO (T2, RC2, NOBAS, L)
      IF (MAXMIN.GT.0) THEN
              RCSUP(IBAS(K)) = -A(MB, IBAS(K)) + RC1
              RCINF(IBAS(K)) = -A(MB, IBAS(K)) - RC2
      ELSE
              RCSUP(IBAS(K)) = A(MB, IBAS(K)) + RC2
              RCINF(IBAS(K)) = A(MB, IBAS(K)) - RC1
      ENDIF
30    CONTINUE
      DO 40 J = 1, NOBAS
      IF (INBAS(J).GT.NO) GO TO 40
      IF (MAXMIN.GT.0) THEN
              RCSUP(INBAS(J)) = -A(MB, INBAS(J)) + ZJCJ(J)
              RCINF(INBAS(J)) = -T6DCJ
      ELSE
              RCSUP(INBAS(J)) = T6DCJ
              RCINF(INBAS(J)) = A(MB, INBAS(J)) - ZJCJ(J)
      ENDIF
40    CONTINUE
      RETURN
      END

      SUBROUTINE MINIMO (TM, TMIN, N, L)
      DOUBLE PRECISION TM, TMIN
      DIMENSION TM(1)
      L = 0
      DO 10 I = 1, N
      IF (TMIN.LE.TM(I)) GO TO 10
      TMIN = TM(I)
      L = I
```

```
10        CONTINUE
          RETURN
          END

          SUBROUTINE ADESEN
          IMPLICIT LOGICAL (S)
          DOUBLE PRECISION A, B
          COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
          COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
          COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
          CHARACTER*12 STATUS
          CHARACTER*8 MODELO
          IF (.NOT.SINSTR) THEN
                  CALL TEXTOS ('ADESMAIN.TXT', 0)
                  CALL PAUSA
                  CALL ALTO (LIN)
          ENDIF
10        CALL MENUS (5)
          WRITE (*, 1001)
1001      FORMAT('0',24X,'( Que opcion eliges ?  '\)
          READ (*, '(A1)', ERR = 10) IRESP
          IF (IRESP.EQ.0) RETURN
          IF (IRESP.EQ.1) CALL ADSEN1
          IF (IRESP.EQ.2) CALL ADSEN2
          IF (IRESP.EQ.3) THEN
                  IF (NO.GE.100) THEN
                          CALL TEXTOS ('ERROR1.TXT  ', 0)
                          CALL PAUSA
                          CALL ALTO (LIN)
                  ELSE
                          CALL ADSEN3
                  ENDIF
          ENDIF
          IF (IRESP.EQ.4) THEN
                  IF (MO.GE.100) THEN
                          CALL TEXTOS ('ERROR1.TXT  ', 0)
                          CALL PAUSA
                          CALL ALTO (LIN)
                  ELSE
                          CALL ADSEN4
                  ENDIF
          ENDIF
          GO TO 10
          END

          SUBROUTINE ADSEN1
          IMPLICIT DOUBLE PRECISION (R, T, X, Y, Z), LOGICAL (S)
          DOUBLE PRECISION A, B, BI, BNUEVO
          COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
          COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
          COMMON /RANGOS/ RBSUP(100), RBINF(100), RCSUP(102), RCINF(102)
          COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
          COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
          COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
          COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
```

```
      COMMON /FORM/ IMP(8), LUN, FMT
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      CHARACTER*8 BNOM, XNOM, ZNOM, FMT, MODELO
      CHARACTER*12 STATUS
      CHARACTER*79 TITULO
      CHARACTER IMP, RESP
      IF (.NOT.SINSTR) THEN
              CALL TEXTOS ('ADES1.TXT    ', 0)
              CALL PAUSA
              CALL ALTO (LIN)
      ENDIF
      CALL VITICS (0)
      WRITE (*, 1001)
      READ (*, '(A1)') RESP
      IF (RESP.EQ.' '.OR.RESP.EQ.'N'.OR.RESP.EQ.'n') GO TO 10
      CALL MENURES
10    CALL LISTR
20    WRITE (*, 1002)
      READ (*, '(I3)', ERR = 20) K
      IF (K.LT.0.OR.K.GT.MO) GO TO 20
      IF (K.EQ.0) RETURN
30    CALL VITICS (0)
      WRITE (*, 1003) K, BNOM(K), B(K)
      READ (*, *, ERR = 30) BNUEVO
      IF (BNUEVO.LT.0.0D0) GO TO 30
      XDELTA = BNUEVO - B(K)
      DO 40 I = 1, MB
40    XB(I) = XB(I) + XDELTA * BI(I, K)
      CALL DUALSI
      IF (SNFACT) THEN
              CALL TEXTOS ('ADES1NOF.TXT', 1)
              CALL PAUSA
              CALL ALTO (LIN)
              DO 50 I = 1, MB
50            XB(I) = XB(I) - XDELTA * BI(I, K)
              CALL DUALSI
              GO TO 10
      ENDIF
      B(K) = B(K) + XDELTA
      IF (ITER1.GT.0) THEN
              CALL RANGOB
              CALL RANGOC
      ENDIF
      METODO = 2
      CALL ALTO (LIN)
      CALL COSTOS
      CALL IMPRIM
      CALL COSTOS
      GO TO 10
1001  FORMAT (///////'0',10X,
     * '( Deseas cambiar los resultados a imprimir ?  '\)
1002  FORMAT (/' ',10X,
     * 'Numero del lado derecho a cambiar [ 0 = ninguno ] :  '\)
1003  FORMAT (/////'0',
     * 10X,'Lado derecho de la restriccion :    ',I3,' .-   ',A8,
```

```
     *  /'0',25X,1P,   'Valor actual   :   ',G18.11,
     *  /' ',25X,      'Nuevo valor    :          '\)
        END

        SUBROUTINE ADSEN2
        IMPLICIT DOUBLE PRECISION (R, T, X, Y, Z), LOGICAL (S)
        DOUBLE PRECISION A, B, BI, CNUEVO
        COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
        COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
        COMMON /VECTDP/ T1(102), T2(102)
        COMMON /RANGOS/ RBSUP(100), RBINF(100), RCSUP(102), RCINF(102)
        COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
        COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
        COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
        COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
        COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
        COMMON /FORM/ IMP(8), LUN, FMT
        COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
        CHARACTER*8 BNOM, XNOM, ZNOM, FMT, MODELO
        CHARACTER*12 STATUS
        CHARACTER*79 TITULO
        CHARACTER IMP, RESP
        CHARACTER*2 FASE
        IF (.NOT.SINSTR) THEN
                CALL TEXTOS ('ADES2.TXT    ', 0)
                CALL PAUSA
                CALL ALTO (LIN)
        ENDIF
        CALL VITICS (0)
        WRITE (*, 1001)
        READ (*, '(A1)') RESP
        IF (RESP.EQ.' '.OR.RESP.EQ.'N'.OR.RESP.EQ.'n')  GO TO 10
        CALL MENURES
10      CALL COSTOS
        CALL FUNOBJ
        CALL COSTOS
20      WRITE (*, 1002)
        READ (*, '(I3)', ERR = 20) K
        IF (K.LT.0.OR.K.GT.NO) GO TO 20
        IF (K.EQ.0) RETURN
        SBASIC = .FALSE.
        DO 30 I = 1, M
        IF (IBAS(I).EQ.K) THEN
                K1 = I
                SBASIC = .TRUE.
                GO TO 40
        ENDIF
30      CONTINUE
40      CALL VITICS (0)
        XDELTA = A(MB, K)
        IF (MAXMIN.GT.0) XDELTA = -XDELTA
        WRITE (*, 1003) K, XNOM(K), XDELTA
        READ (*, *, ERR = 40) CNUEVO
        XDELTA = CNUEVO - XDELTA
        IF (MAXMIN.GT.0) THEN
```

```
                A(MB, K) = A(MB, K) - XDELTA
        ELSE
                A(MB, K) = A(MB, K) + XDELTA
        ENDIF
        IF (SBASIC) THEN
                DO 60 I = 1, M
60              BI(MB, I) = BI(MB, I) + XDELTA * BI(K1, I)
                XB(MB) = XB(MB) + XDELTA * XB(K1)
        ENDIF
        MFASE = MB
        FASE = 'II'
        NFREC = 0
        CALL MENUS (8)
        CALL SIMREV (NFREC, FASE, MFASE, L, ITER)
        IF (SNACOT) THEN
                CALL TEXTOS ('ADES2NOA.TXT', 0)
                WRITE (*, 1004) L, XNOM(L)
                CALL PAUSA
                CALL ALTO (LIN)
                IF (MAXMIN.GT.0) THEN
                        A(MB, K) = A(MB, K) + XDELTA
                ELSE
                        A(MB, K) = A(MB, K) - XDELTA
                ENDIF
                IF (SBASIC) THEN
                        DO 70 I = 1, M
70                      BI(MB, I) = BI(MB, I) - XDELTA * BI(K1, I)
                        XB(MB) = XB(MB) - XDELTA * XB(K1)
                ENDIF
                CALL SIMREV (NFREC, FASE, MFASE, L, ITER)
                GO TO 10
        ENDIF
        WRITE (*, 1005)
        IF (ITER.GT.0) THEN
                CALL RANGOB
                CALL RANGOC
        ENDIF
        CALL ALTO (LIN)
        METODO = 1
        ITER1 = 0
        ITER2 = ITER
        CALL COSTOS
        CALL IMPRIM
        CALL COSTOS
        GO TO 10
1001    FORMAT (///////'0',10X,
     *  '( Deseas cambiar los resultados a imprimir ?   '\)
1002    FORMAT (/' ',10X,
     *  'Numero del costo a cambiar [ 0 = ninguno ] :   '\)
1003    FORMAT (/////'0',
     *  10X,'Variable  X (',I3,' )    ',A8,
     *  /'0',15X,1P,  'Contribucion actual  :  ',G18.11,
     *  /' ',15X,      'Nueva contribucion     :            '\)
1004    FORMAT (' ',20X,'La variable no acotada es X(',I2,')   ',A8,//)
1005    FORMAT (//'0',31X,      'Optimo alcanzado.'///
```

```
     *              ' ',23X,'Teclea [ RETURN ] para continuar.   '\)
      END

      SUBROUTINE ADSEN3
      IMPLICIT DOUBLE PRECISION (T, X, Y, Z), LOGICAL (S)
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      DOUBLE PRECISION A, B, BI
      COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
      COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      COMMON /FORM/ IMP(8), LUN, FMT
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      CHARACTER*8 BNOM, XNOM, ZNOM, FMT, MODELO
      CHARACTER*12 STATUS
      CHARACTER*79 TITULO
      CHARACTER IMP, RESP
      CHARACTER*2 FASE
      SOLOPT = .FALSE.
      IF (.NOT.SINSTR) THEN
              CALL TEXTOS ('ADES3.TXT   ', 0)
              CALL PAUSA
              CALL ALTO (LIN)
      ENDIF
      CALL VITICS (0)
      WRITE (*, 1000)
      READ (*, '(A1)') RESP
      IF (RESP.EQ.' '.OR.RESP.EQ.'N'.OR.RESP.EQ.'n')  GO TO 10
      CALL MENURES
 10   K1 = NO + 1
      CALL VITICS (0)
      WRITE (*, 1001)
      WRITE (*, 1002) K1
      READ (*, '(A8)') XNOM(K1)
      CALL VITICS (0)
      WRITE (*, 1003) K1
      CALL ALTO (LIN)
 20   CALL VITICS (0)
      WRITE (*, 1004) K1, XNOM(K1)
      DO 30 I = 1, MO
 25   WRITE (*, 1005) I, BNOM(I), K1
 30   READ (*, *, ERR = 25) A(I, K1)
      WRITE (*, 1004) K1, XNOM(K1)
 35   WRITE (*, 1007) K1, XNOM(K1)
      READ (*, *, ERR = 35) A(MB, K1)
      IF (MAXMIN.GT.0) A(MB, K1) = -A(MB, K1)
      WRITE (*, 1008)
      READ (*, '(A1)') RESP
      IF (RESP.EQ.'S'.OR.RESP.EQ.'s') GO TO 20
      DO 40 I = 1, MB
 40   IF (IBAS(I).GT.NO) IBAS(I) = IBAS(I) + 1
      DO 50 J = 1, NOBAS
 50   IF (INBAS(J).GT.NO) INBAS(J) = INBAS(J) + 1
      NO = NO + 1
```

```
      NOH = NOH + 1
      NOBAS = NOBAS + 1
      NOHA = NOHA + 1
      INBAS(NOBAS) = K1
      CALL MENUS (8)
      NFREC = 0
      FASE = 'II'
      MFASE = MB
      CALL SIMREV (NFREC, FASE, MFASE, K, ITER)
      IF (SNACOT) THEN
              CALL TEXTOS ('ADES3NOA.TXT', 0)
              WRITE (*, 1009) K, XNOM(K)
              CALL PAUSA
              CALL ALTO (LIN)
              RETURN
      ENDIF
      SOLOPT = .TRUE.
      WRITE (*, 1011)
      IF (ITER.GT.0) THEN
              CALL RANGOB
              CALL RANGOC
      ENDIF
      CALL ALTO (LIN)
      ITER1 = 0
      ITER2 = ITER
      METODO = 1
      CALL COSTOS
      CALL IMPRIM
      CALL COSTOS
      CALL VITICS (0)
      WRITE (*, 1010)
      READ (*, '(A1)') RESP
      IF (RESP.EQ.'S'.OR.RESP.EQ.'s') GO TO 10
      RETURN
1000  FORMAT (///////'0',10X,
     * '( Deseas cambiar los resultados a imprimir ?  '\)
1001  FORMAT (////
     * '0',10X,'( Cu ì es el nombre de la nueva variable ?'//)
1002  FORMAT (' ',15X,'Variable',I4,7X,'Nombre :   '\)
1003  FORMAT (////'0',/
     *' ',10x,'  Bien.  Dame por favor  los  coeficientes de lá nueva'/
     *' ',10X,'variable  X(',I3,' )  en cada una de las restricciones'/
     *' ',10X,'originales.'/
     *'0',12X,'Teclea [ RETURN ] para empezar.   '\)
1004  FORMAT ('0',21X,'******  Variable',I4,' ******'/' ',32X,A8,/)
1005  FORMAT (/'+',5X,
     * 'Restriccion',I4,2X,A8,4X,'Coeficiente de X(',I3,' ) : '\)
1007  FORMAT(/' ',5X,
     *'Contribucion de X(',I3,' ) ',A8,' a la funcion objetivo : '\)
1008  FORMAT ('0',10X,'( Deseas hacer alguna correccion [ S/N ] ?   '\)
1009  FORMAT (' ',20X,'La variable no acotada es X(',I2,')    ',A8,//)
1010  FORMAT (///////'0',20X,'( Deseas agregar otra variable ?  '\)
1011  FORMAT (//'0',30X,      'Optimo alcanzado.'///
     *             ' ',23X,'Teclea [ RETURN ] para continuar.   '\)
      END
```

```
      SUBROUTINE ADSEN4
      IMPLICIT DOUBLE PRECISION (T, X, Y, Z), LOGICAL (S)
      DOUBLE PRECISION A, B, BI
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
      COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
      COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      COMMON /FORM/ IMP(8), LUN, FMT
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      CHARACTER IMP, RESP
      CHARACTER*2 FASE
      CHARACTER*8 BNOM, XNOM, ZNOM, FMT, MODELO
      CHARACTER*12 STATUS
      CHARACTER*79 TITULO
      SOLOPT = .FALSE.
      IF (.NOT.SINSTR) THEN
              CALL TEXTOS ('ADES4.TXT    ', 0)
              CALL PAUSA
              CALL ALTO (LIN)
      ENDIF
      CALL VITICS (0)
      WRITE (*, 1000)
      READ (*, '(A1)') RESP
      IF (RESP.EQ.' '.OR.RESP.EQ.'N'.OR.RESP.EQ.'n') GO TO 10
      CALL MENURES
10    MB = MB + 1
      M = M  + 1
      MO = M
      B(MB) = B(M)
      MRESTR(MB) = MRESTR(M)
      XB(MB) = XB(M)
      BI(MB, MB) = 1.0D0
      IBAS(MB) = IBAS(M) + 1
      DO 20 J = 1, NO
20    A(MB, J) = A(M, J)
      CALL VITICS (0)
      WRITE (*, 1001)
      WRITE (*, 1002) M
      READ (*, '(A8)') BNOM(M)
      CALL RESTR (M)
      GO TO (30, 40, 50), MRESTR(M)
30    NHPOS = NHPOS + 1
      NH = NH + 1
      NOH = NOH + 1
      NOHA = NOHA + 1
      IHOL(NH, 1) = M
      IHOL(NH, 2) = 1
      IBAS(M) = NOH
      GO TO 60
40    NHNEG = NHNEG + 1
      NH = NHNEG + NHPOS
```

```
      NOH = NH + NO
      NOHA = NOHA + 1
      IHOL(NH, 1) = M
      IHOL(NH, 2) = -1
      IBAS(M) = NOH
      BI(M, M) = -1.0D0
      GO TO 60
 50   NRE = NRE + 1
      NA = NA + 1
      NOHA = NOHA + 1
      IBAS(M) = NOHA
      IHOL(NH + NA, 1) = M
      IHOL(NH + NA, 2) = 1
 60   M1 = M - 1
      DO 70 J = 1, M1
      BI(MB, J) = BI(M, J)
 70   BI(M, J) = 0.0D0
      BI(MB, M) = 0.0D0
      DO 80 I = 1, M1
      DO 80 J = 1, M1
 80   BI(M, I) = BI(M, I) - A(M, IBAS(J)) * BI(J, I)
      IF (MRESTR(M).EQ.2) THEN
              DO 90 J = 1, M1
 90           BI(M, J) = -BI(M, J)
      ENDIF
      DO 100 I = 1, M
      XB(I) = 0.0D0
      DO 100 J = 1, M
 100  XB(I) = XB(I) + BI(I, J) * B(J)
      IF (MRESTR(M).EQ.3.AND.XB(M).GE.T4SNF) THEN
              PRECIO = T6DCJ / 100.
              DO 110 J = 1, M1
 110          BI(MB, J) = BI(MB, J) - PRECIO * BI(M, J)
              BI(MB, M) = -PRECIO
              XB(MB) = 0.0D0
              DO 120 J = 1, MB
 120          XB(MB) = XB(MB) + BI(MB, J) * B(J)
              MFASE = MB
              FASE = 'II'
              NFREC = 0
              CALL MENUS (8)
              CALL SIMREV (NFREC, FASE, MFASE, L, ITER)
              IF (IBAS(M).EQ.NOHA.AND.XB(M).GT.-T4SNF) THEN
                      CALL TEXTOS ('ADES4NOF.TXT', 0)
                      CALL PAUSA
                      CALL ALTO (LIN)
                      RETURN
              ENDIF
              WRITE (*, 1004)
              ITER2 = ITER
              METODO = 1
      ELSE
              BI(MB, M) = 0.0D0
              CALL DUALSI
              IF (SNFACT) THEN
```

```
                            CALL TEXTOS ('ADES4NOF.TXT', 0)
                            CALL PAUSA
                            CALL ALTO (LIN)
                            RETURN
                     ENDIF
                     METODO = 2
              ENDIF
              SOLOPT = .TRUE.
              IF (ITER1.GT.0) THEN
                     CALL RANGOB
                     CALL RANGOC
              ENDIF
              CALL ALTO (LIN)
              CALL COSTOS
              CALL IMPRIM
              CALL COSTOS
              CALL VITICS (0)
              WRITE (*, 1003)
              READ (*, '(A1)') RESP
              IF (RESP.EQ.'S'.OR.RESP.EQ.'s') GO TO 10
              RETURN
 1000         FORMAT (///////'0',10X,
      *       '( Deseas cambiar los resultados a imprimir ?  '\)
 1001         FORMAT (
      *       ////'0',10X,'( Cu 1 es el nombre de la nueva restriccion ?'//)
 1002         FORMAT (' ',12X,'Restriccion',I4,7X,'Nombre :  '\)
 1003         FORMAT (///////'0',20X,'( Deseas agregar otra restriccion ?  '\)
 1004         FORMAT (//'0',30X,        'Optimo alcanzado.'///
      *                 ' ',23X,'Teclee [ RETURN ] para continuar.  '\)
              END

              SUBROUTINE REPORT
              IMPLICIT LOGICAL (S)
              COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
              COMMON /FORM/ IMP(8), LUN, FMT
              COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
              CHARACTER IMP, RESP
              CHARACTER*8 FMT, MODELO
              CHARACTER*12 STATUS
              IF (.NOT.SINSTR) THEN
                     CALL TEXTOS ('REPORT.TXT   ', 0)
                     CALL PAUSA
                     CALL ALTO (LIN)
              ENDIF
 10           CALL MENUS (4)
              WRITE (*, 1001)
 1001         FORMAT ('0',26X,'( Que opcion eliges ?  '\)
              READ (*, '(A1)') RESP
              IF (RESP.EQ.' ') RETURN
              IF (RESP.EQ.'P'.OR.RESP.EQ.'p'.OR.RESP.EQ.'I'.OR.RESP.EQ.'i'
      *       .OR.RESP.EQ.'D'.OR.RESP.EQ.'d') CALL UNIDAD (RESP)
              IF (RESP.EQ.'R'.OR.RESP.EQ.'r') CALL MENURES
              IF (RESP.EQ.'E'.OR.RESP.EQ.'e') CALL IMPRIM
              GO TO 10
              END
```

```
      SUBROUTINE UNIDAD (RESP)
      COMMON /FORM/ IMP(8), LUN, FMT
      CHARACTER IMP, RESP
      CHARACTER*8 FMT
      IF (RESP.EQ.'I'.OR.RESP.EQ.'i') THEN
              LUN = -1
              CALL VITICS (0)
              WRITE (*, 1000)
              CALL ALTO (LIN)
      ELSEIF (RESP.EQ.'D'.OR.RESP.EQ.'d') THEN
              LUN = 0
              CALL VITICS (0)
              WRITE (*, 1001)
              CALL ALTO (LIN)
      ELSEIF (RESP.EQ.'P'.OR.RESP.EQ.'p') THEN
              LUN = 1
      ENDIF
      RETURN
1000  FORMAT (//////////'0',17X,
     *        'Por favor cerci"rate de que la impresora este'/
     *' ',15X,'ON LINE  y teclea  [ RETURN ]  para continuar. '\)
1001  FORMAT (//////////'0',17X,
     *        'Por favor cerci"rate de que  haya  un'/
     *' ',15X,'diskette en el drive que vas a utilizar'/
     *' ',15X,'y teclea  [ RETURN ]  para continuar. '\)
      END

      SUBROUTINE MENURES
      IMPLICIT LOGICAL (S)
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      COMMON /FORM/ IMP(8), LUN, FMT
      CHARACTER IMP, RESP
      CHARACTER*8 FMT, MODELO
      CHARACTER*12 STATUS
      IF (.NOT.SINSTR) THEN
              CALL TEXTOS ('MENURES1.TXT', 0)
              CALL PAUSA
              CALL ALTO (LIN)
      ENDIF
      CALL MENUS (6)
      WRITE (*, 1001)
      READ (*, '(A1)') RESP
      IF (RESP.EQ.' '.OR.RESP.EQ.'S'.OR.RESP.EQ.'s') THEN
              CALL SELECT ('*', '*', '*', '*', '*', '*', '*', '*')
              RETURN
      ENDIF
10    CALL SELECT (' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ')
      CALL MENUS (7)
      CALL VITICS (1)
      DO 20 I = 1, 8
      WRITE (*, 1002) I
      READ (*, '(A1)') RESP
      IF (RESP.EQ.'N'.OR.RESP.EQ.'n'.OR.RESP.EQ.' ') THEN
              IMP(I) = ' '
```

```
        ELSE
                IMP(I) = '*'
        ENDIF
20      CONTINUE
        CALL MENUS (6)
        WRITE (*, 1003)
        READ (*, '(A1)') RESP
        IF (RESP.EQ.'S'.OR.RESP.EQ.'s') GO TO 10
        RETURN
1001    FORMAT (' ',17X,  '( Deseas todas las opciones [ S/N ] ?  '\)
1002    FORMAT (' ',19X,  '( Deseas la opcion [',I1,'], ( S/N ) ?  '\)
1003    FORMAT ('0',21X,  '( Deseas hacer algun cambio ?  '\)
        END

        SUBROUTINE LISTPL
        COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
        COMMON /FORM/ IMP(8), LUN, FMT
        CHARACTER IMP
        CHARACTER*8 FMT
        FMT = 'LISTADOS'
        CALL FUNOBJ
        IF (LUN.GT.0) CALL ALTO (LIN)
        DO 10 I = 1, MO
10      CALL LISTV (I)
        IF (LUN.GT.0) THEN
                OPEN (2, FILE = 'CON')
        ELSE
                OPEN (2, FILE = 'LPT1')
        ENDIF
        WRITE (2, 1000)
1000    FORMAT (//'0',20X,'Todas las variables no negativas.'/////)
        CLOSE (2)
        IF (LUN.GT.0) CALL ALTO (LIN)
        FMT = '        '
        RETURN
        END

        SUBROUTINE SELECT (A1, A2, A3, A4, A5, A6, A7, A8)
        IMPLICIT CHARACTER (A)
        COMMON /FORM/ IMP(8), LUN, FMT
        CHARACTER*8 FMT
        CHARACTER IMP
        DO 10 I = 1, 8
10      IMP(I) = ' '
        IMP(1) = A1
        IMP(2) = A2
        IMP(3) = A3
        IMP(4) = A4
        IMP(5) = A5
        IMP(6) = A6
        IMP(7) = A7
        IMP(8) = A8
        RETURN
        END
```

```
      SUBROUTINE ORDEN (LE, LS, N)
      DIMENSION LE(1), LS(1)
      LMAX = 0
      DO 20 J = 1, N
      LMIN = 700
      DO 10 I = 1, N
      IF (LE(I).GT.LMIN.OR.LE(I).LE.LMAX) GO TO 10
      LMIN = LE(I)
      L = I
10    CONTINUE
      LS(J) = L
      LMAX = LE(L)
20    CONTINUE
      RETURN
      END

      SUBROUTINE IMPRIM
      IMPLICIT DOUBLE PRECISION (R, T, X, Y, Z), LOGICAL (S)
      DOUBLE PRECISION A, B, BI, PRSOMB
      COMMON /DATOS/ A(102, 100), B(102), MRESTR(102), M, MB, NO
      COMMON /COMPDP/ BI(102, 102), XB(102), ZJCJ(200)
      COMMON /RANGOS/ RBSUP(100), RBINF(100), RCSUP(102), RCINF(102)
      COMMON /PARAM/ T1ZKCK, T2XBYJ, T3YJO, T4SNF, T5DBI, T6DCJ
      COMMON /INDICS/ IBAS(102), INBAS(200), IHOL(202, 2)
      COMMON /CTESP/ NH, NA, NOH, NOHA, MO, NOBAS, MAXMIN
      COMMON /CTESS/ NHPOS, NHNEG, NRE, ITER1, ITER2, METODO
      COMMON /NOMBRS/ XNOM(100), BNOM(102), ZNOM, TITULO(3)
      COMMON /SOLUCN/ SINSTR, SOLOPT, SNFACT, SNACOT, MODELO, STATUS
      COMMON /FORM/ IMP(8), LUN, FMT
      COMMON /INDAUX/ LB(102), LNB(102)
      LOGICAL ARCHIV
      CHARACTER IMP, RESP, PAUSSA, HOLGR
      CHARACTER*4 D
      CHARACTER*8 BNOM, XNOM, ZNOM, FMT, MODELO
      CHARACTER*9 ILIM
      CHARACTER*12 STATUS
      CHARACTER*14 LPNOM
      CHARACTER*79 TITULO
5     CALL VITICS (0)
      IF (LUN.EQ.-1) THEN
              WRITE (*, 1003)
      ELSEIF (LUN.EQ.0) THEN
              WRITE (*, 998) MODELO
              READ (*, '(A14)') LPNOM
              IF (LPNOM.EQ.'              ') RETURN
              INQUIRE (FILE = LPNOM, EXIST = ARCHIV)
              IF (ARCHIV) THEN
                      WRITE (*, 999) LPNOM
                      READ (*, '(A1)') RESP
                      IF (RESP.NE.'S'.AND.RESP.NE.'s'.AND.RESP.NE.' ')
     *                    GO TO 5
              ENDIF
              WRITE (*, 1000)
      ENDIF
      CALL ORDEN (IBAS, LB, MB)
```

```
      CALL ORDEN (INBAS, LNB, NOBAS)
      IF (LUN.EQ.1) THEN
              OPEN (2, FILE='CON')
      ELSEIF (LUN.EQ.0) THEN
              OPEN (2, FILE = LPNOM, STATUS = 'NEW')
      ELSEIF (LUN.EQ.-1) THEN
              OPEN (2, FILE='LPT1')
      ENDIF
      DO 10 I = 1, 3
10    WRITE (2, 1001) TITULO(I)
      CLOSE (2)
      IF (IMP(8).EQ.'*') CALL LISTPL
      IF (.NOT.SOLOPT) RETURN
      IF (LUN.EQ.1) THEN
              OPEN (2, FILE='CON')
      ELSEIF (LUN.EQ.0) THEN
              OPEN (2, FILE = LPNOM, STATUS = 'NEW')
      ELSEIF (LUN.EQ.-1) THEN
              OPEN (2, FILE='LPT1')
      ENDIF
      IF (IMP(1).EQ.'*') THEN
              WRITE (2, 1004)
              IF (METODO.EQ.1) WRITE (2, 1051) ITER1, ITER2
              IF (METODO.EQ.2) WRITE (2, 1052) ITER1
              IF (MAXMIN.GT.0)  THEN
                      D = ' MAX'
              ELSE
                      D = ' MIN'
              ENDIF
              WRITE (2, 1006)
              WRITE (2, 1007) D, ZNOM, XB(MB)
              IF (LUN.GT.0) READ (*, '(A1)') PAUSSA
      ENDIF
      IF (IMP(2).EQ.'*') THEN
              WRITE (2, 1008)
              LIN = 0
              DO 20 I = 1, M
              IB = IBAS(LB(I))
              IF (IB.GT.NO) GO TO 20
              WRITE (2, 1009) IB, XNOM(IB), XB(LB(I))
              LIN = LIN + 1
              IF (LIN.EQ.20.AND.LUN.GT.0) THEN
                      READ (*, '(A1)') PAUSSA
                      LIN = 0
              ENDIF
20            CONTINUE
              IF (LUN.GT.0) READ (*, '(A1)') PAUSSA
      ENDIF
      IF (IMP(3).EQ.'*') THEN
              WRITE (2, 1011)
              NH = NOH - NO
              LIN = 0
              DO 30 I = 1, M
              IB = IBAS(LB(I)) - NO
              IF (IB.LE.0.OR.IB.GT.NH) GO TO 30
```

```
                IH = IHOL(IB, 1)
                IF (MRESTR(IH).EQ.1) THEN
                        HOLGR = '+'
                ELSE
                        HOLGR = '-'
                ENDIF
                WRITE (2, 1014) IH, HOLGR, BNOM(IH), XB(LB(I))
                LIN = LIN + 1
                IF (LIN.EQ.20.AND.LUN.GT.0) THEN
                        READ (*, '(A1)') PAUSSA
                        LIN = 0
                ENDIF
30              CONTINUE
                IF (LUN.GT.0) READ (*, '(A1)') PAUSSA
        ENDIF
        IF (IMP(4).EQ.'*') THEN
                WRITE (2, 1012)
                LIN = 0
                DO 40 I = 1, MO
                IF (DABS(BI(MB, I)).LT.-T1ZKCK) GO TO 40
                WRITE (2, 1010) I, BNOM(I), BI(MB, I)
                LIN = LIN + 1
                IF (LIN.EQ.20.AND.LUN.GT.0) THEN
                        READ (*, '(A1)') PAUSSA
                        LIN = 0
                ENDIF
40              CONTINUE
                IF (LUN.GT.0) READ (*, '(A1)') PAUSSA
        ENDIF
        IF (IMP(5).EQ.'*') THEN
                WRITE (2, 1013)
                LIN = 0
                DO 50 J = 1, NOBAS
                INB = INBAS(LNB(J))
                IF (INB.GT.NO) GO TO 50
                REDCST = -ZJCJ(LNB(J))
                IF (DABS(REDCST).LT.-T1ZKCK) GO TO 50
                WRITE (2, 1010) INB, XNOM(INB), REDCST
                LIN = LIN + 1
                IF (LIN.EQ.20.AND.LUN.GT.0) THEN
                        READ (*, '(A1)') PAUSSA
                        LIN = 0
                ENDIF
50              CONTINUE
                IF (LUN.GT.0) READ (*, '(A1)') PAUSSA
        ENDIF
        IF (IMP(6).EQ.'*') THEN
                ILIM = 'Ilimitado'
                WRITE (2, 1018)
                LIN = 0
                DO 60 J = 1, NO
                  IF (2.0D0 * DABS(RCINF(J)).GT.T6DCJ.AND.
     *        2.0D0 * DABS(RCSUP(J)).GT.T6DCJ) THEN
                WRITE (2, 1017) J, XNOM(J), ILIM, A(MB, J), ILIM
                  ELSEIF (2.0D0 * DABS(RCINF(J)).GT.T6DCJ) THEN
```

```
              WRITE (2, 2017) J, XNOM(J), ILIM, A(MB, J), RCSUF(J)
                ELSEIF (2.0D0 * DABS(RCSUF(J)).GT.T6DCJ) THEN
              WRITE (2, 3017) J, XNOM(J), RCINF(J), A(MB, J), ILIM
                ELSE
              WRITE (2, 4017) J, XNOM(J), RCINF(J), A(MB, J), RCSUF(J)
                ENDIF
              LIN = LIN + 1
              IF (LIN.EQ.20.AND.LUN.GT.0) THEN
                      READ (*, '(A1)') PAUSSA
                      LIN = 0
              ENDIF
60            CONTINUE
              IF (LUN.GT.0) READ (*, '(A1)') PAUSSA
      ENDIF
      IF (IMP(7).EQ.'*') THEN
              ILIM = 'Ilimitado'
              WRITE (2, 1019)
              LIN = 0
              DO 70 I = 1, MO
        IF (2.0D0 * DABS(RBINF(I)).GT.T5DBI.AND.
     *        2.0D0 * DABS(RBSUP(I)).GT.T5DBI) THEN
              WRITE (2, 1016) I, BNOM(I), ILIM, B(I), ILIM
        ELSEIF (2.0D0 * DABS(RBINF(I)).GT.T5DBI) THEN
              WRITE (2, 2016) I, BNOM(I), ILIM, B(I), RBSUP(I)
        ELSEIF (2.0D0 * DABS(RBSUP(I)).GT.T5DBI) THEN
              WRITE (2, 3016) I, BNOM(I), RBINF(I), B(I), ILIM
        ELSE
              WRITE (2, 4016) I, BNOM(I), RBINF(I), B(I), RBSUP(I)
        ENDIF
              LIN = LIN + 1
              IF (LIN.EQ.20.AND.LUN.GT.0) THEN
                      READ (*, '(A1)') PAUSSA
                      LIN = 0
              ENDIF
70            CONTINUE
              IF (LUN.GT.0) READ (*, '(A1)') PAUSSA
      ENDIF
      CLOSE (2)
      RETURN
998   FORMAT (///////
     *  '0',12X, '( En que archivo deseas grabar el reporte'/
     *  ' ',10X,'de resultados del Modelo ',A8,' ?'/
     *  '0',10X,'Nombre del archivo [ A:NOMBRE.REP ] :  '\)
999   FORMAT ('0',10X,
     *  '( Deseas borrar el antiguo archivo ',A14,'?  '\)
1000  FORMAT (/'0',25X,'Grabando ...   '\)
1001  FORMAT (/' ',A79)
1002  FORMAT ('  ',/)
1003  FORMAT (////////'0',25X,'Imprimiendo ...'//)
1004  FORMAT (///' ',5X,'SOLUCION OPTIMA ENCONTRADA DESPUES DE'/)
1051  FORMAT (' ', 8X,       I5,'  ITERACIONES DE FASE  I'
     *     /' ', 8X,       I5,'  ITERACIONES DE FASE  II'
     *    //' ',10X,      'DEL METODO SIMPLEX REVISADO.'//)
1052  FORMAT (' ',10X,        I2,'  ITERACIONES'
     *    //' ',10X,    'DEL METODO DUAL SIMPLEX.'//)
```

```
1006     FORMAT (/' ',5X,
     *   'LOS RESULTADOS NO LISTADOS EN LAS SECCIONES 2-5 VALEN CERO.'//)
1007     FORMAT (/' ',5X,
     *     '1 -  VALOR',A4,'IMO DE LA FUNCION OBJETIVO',
     *   //' ',1P,10X,A8,10X,G18.8,5X,\)
1008     FORMAT(///' ',5X,
     *   '2 -  VALORES OPTIMOS DE LAS VARIABLES DE DECISION'/)
1009     FORMAT (' ',1P,10X,'X (',I3,' )',5X,A8,7X,G18.8)
1010     FORMAT (' ',1P,10X,I3,'.-  ',A8,7X,G18.8)
1011     FORMAT(///' ',5X,
     *   '3 -  HOLGURA (+) Y EXCESO (-) EN LAS RESTRICCIONES'/)
1012     FORMAT(///' ',5X,
     *   '4 -  VALORES DE LAS VARIABLES DUALES'/)
1013     FORMAT(///' ',5X,
     *   '5 -  COSTOS REDUCIDOS DE LAS VARIABLES DE DECISION'/)
1014     FORMAT (' ',10X,I3,'.-  ',1P,A1,1X,A8,7X,G18.8)
1016     FORMAT (' ',I3,'.-   ',1P,A8,8X,A9,7X,G16.6,8X,A9)
2016     FORMAT (' ',I3,'.-   ',1P,A8,8X,A9,7X,G16.6,4X,G16.6)
3016     FORMAT (' ',I3,'.-   ',1P,A8,4X,G16.6,4X,G16.6,8X,A9)
4016     FORMAT (' ',I3,'.-   ',1P,A8,4X,G16.6,4X,G16.6,4X,G16.6)
1017     FORMAT (' X (',I3,' )   ',1P,A8,8X,A9,7X,G16.6,8X,A9)
2017     FORMAT (' X (',I3,' )   ',1P,A8,8X,A9,7X,G16.6,4X,G16.6)
3017     FORMAT (' X (',I3,' )   ',1P,A8,4X,G16.6,4X,G16.6,8X,A9)
4017     FORMAT (' X (',I3,' )   ',1P,A8,4X,G16.6,4X,G16.6,4X,G16.6)
1018     FORMAT (///' ',5X,
     *   '6 -  RANGOS DE VARIACION DE LOS COSTOS ',
     *                              'DE LA FUNCION OBJETIVO'//
     *   ' ',5X,'VARIABLE',9X,'LIMITE INFERIOR',9X,'ACTUAL',9X,
     *                                                  'LIMITE SUPERIOR'/
     *   ' ',5X,'--------',9X,'---------------',9X,'------',9X,
     *                                                  '---------------'/)
1019     FORMAT (///' ',5X,
     *   '7 -  RANGOS DE VARIACION DE LOS LADOS DERECHOS'//
     *   ' ',5X,'RESTRICCION',5X,'LIMITE INFERIOR',9X,'ACTUAL',9X,
     *                                                  'LIMITE SUPERIOR'/
     *   ' ',5X,'-----------',5X,'---------------',9X,'------',9X,
     *                                                  '---------------'/)
         END
```