CONTENIDO

FIGURAS

TABLAS

2.1 Programación lineal (PL)

Un problema de programación lineal es un caso simplificado del problema general de optimización matemática. Consiste en una función objetivo lineal y una o más restricciones lineales [LINDO].

Fue desarrollado para resolver problemas de planeación de la fuerza aérea americana. La programación lineal permite seleccionar los mejores valores de las variables de decisión en situaciones donde las variables compiten con recursos limitados. Un problema de programación lineal se define como:

$$\begin{aligned} &Maximize / Minimize \quad Z = \sum_{j=1}^{n} c_j x_j \\ &Sujeto \quad a: \\ &\sum_{j=1}^{n} a_{ij} x_j \quad \{\leq, \equiv, \geq\} \quad b_i \quad (i = 1, ..., m) \\ &x_j \geq 0 \quad (j = 1, ...n) \end{aligned} \tag{2.1}$$

Donde:

- *n* Número de variables de decisión.
- *m* Número de restricciones.
- *a* Coeficiente conocido de cada variable en las restricciones.
- *b* Valor conocido ubicado en el lado derecho de las restricciones.
- *c* Coeficiente conocido que denota la relación costo/beneficio de la variable de decisión.

2.1.1 Áreas de aplicación:

- Desarrollar calendarios de trabajo que utilicen eficientemente el tiempo del personal.
- Asignar espacios de fabricación que permitan mejorar las líneas de producción.

- Determinar los niveles de gastos de varias actividades.
- Optimizar la mezcla de productos de una operación.
- Asignación del equipo para los vehículos del sistema de transporte publico.
- Problemas clásicos de transporte
    - Problema del cartero
    - Problema del agente viajero
    - Problema de rutas de vehículos
    - Problema del transporte

2.1.2 Casos especiales de programación lineal: Problema de transporte

Este tipo de problemas consisten en minimizar el costo de transporte de unidades desde cierto número de lugares de origen a cierto número de lugares de destino [Anderson 1991].

2.1.2.1 Ejemplo clásico del problema de transporte

Una empresa enfrenta el siguiente problema de transporte: minimizar el costo de transporte de productos desde 3 plantas (P1, P2 y P3) hacia 4 centros de distribución (D1, D2, D3 y D4).

En la Figura 2.1, se muestra la capacidad de producción de cada planta.

Origen	Planta	Capacidad de Producción (unidades)
1	P1	5,000
2	P2	6,000
3	P3	2,500
Total		13,500

Figura 2.1 Capacidad de producción de las plantas de la empresa

En la Figura 2.2, se muestra las demandas respectivas de los centros de distribución:

Destino	Centro de distribución	Demanda (unidades)
1	D1	6,000
2	D2	4,000
3	D3	2,000
4	D4	1,500
Total		13,500

Figura 2.2 Demanda los centros de distribución de la empresa

El objetivo es determinar las rutas que deben utilizarse y la cantidad que se debe de enviar a través de cada ruta, de tal manera que el costo total de transporte sea el mínimo y que satisfagan las demandas de todos los centros de distribución. En la Figura 2.3, se presentan los costos de transporte por unidad de producto:

		Destino			
		D1	**D2**	**D3**	**D4**
Origen	**P1**	3	2	7	6
	P2	7	5	2	3
	P3	2	5	4	5

Figura 2.3 Costos unitarios de transporte

Considerando el planteamiento anterior, se describen las etapas de la representación del problema de transporte:

- **Solución**

Se considera x_{ijs} como el número de unidades que serán transportadas desde la planta i al centro de distribución j.

La función objetivo es:

$$Min \quad \begin{aligned}&(3x_{11}+2x_{12}+7x_{13}+6x_{14})+(7x_{21}+5x_{22}+2x_{23}+3x_{24})\\&+(2x_{31}+5x_{32}+4x_{33}+5x_{34})\end{aligned} \tag{2.2}$$

- Restricciones

1.- No exceder la capacidad de las plantas:

$$\begin{aligned}&x_{11}+x_{12}+x_{13}+x_{14}\le 5,000 \quad .....(1)\\&x_{21}+x_{22}+x_{23}+x_{24}\le 6,000 \quad .....(2)\\&x_{31}+x_{32}+x_{33}+x_{34}\le 2,500 \quad .....(3)\end{aligned} \tag{2.3}$$

2.- Satisfacer la demanda de los centros de distribución:

$$\begin{aligned}&x_{11}+x_{21}+x_{31}\ge 6,000 \quad .....(4)\\&x_{12}+x_{22}+x_{32}\ge 4,000 \quad .....(5)\\&x_{13}+x_{23}+x_{33}\ge 2,000 \quad .....(6)\\&x_{14}+x_{24}+x_{34}\ge 1,500 \quad .....(7)\end{aligned} \tag{2.4}$$

- **El modelo**

La Figura 2.4 muestra el problema de programación lineal resultante.

Min

$$\begin{array}{cccccccccccc}
3x_{11} & +2x_{12} & +7x_{13} & +6x_{14} & +7x_{21} & +5x_{22} & +2x_{23} & +3x_{24} & +2x_{31} & +5x_{32} & +4x_{33} & +5x_{34} & & \\
x_{11} & +x_{12} & +x_{13} & +x_{14} & & & & & & & & & \le & 5,000 \\
 & & & & x_{21} & +x_{22} & +x_{23} & +x_{24} & & & & & \le & 6,000 \\
 & & & & & & & & x_{31} & +x_{32} & +x_{33} & +x_{34} & \le & 2,500 \\
x_{11} & & & & +x_{21} & & & & +x_{31} & & & & = & 6,000 \\
 & x_{12} & & & & +x_{22} & & & & +x_{32} & & & = & 4,000 \\
 & & x_{13} & & & & +x_{23} & & & & +x_{33} & & = & 2,000 \\
 & & & x_{14} & & & & +x_{24} & & & & +x_{34} & = & 1,500
\end{array}$$

$x_{ij} \ge 0$ para $i = 1,2,3$ y j=1,2,3,4

Figura 2.4 Representación del problema

2.1.2.2 Solución automatizada del ejemplo clásico del problema de transporte

Se utiliza el software LINDO (Linear Integer Discrete Optimizer) para resolverlo. En la Figura 2.5 se muestra la representación del problema como la relación entre cantidad a enviar, demanda y capacidad. En la Figura 2.6 se muestra el script proporcionado a LINDO y en la Figura 2.7 obtenemos la información de salida.

Datos de Entrada:

	1	2	3	4	Capacidad
1	?	?	?	?	5,000
2	?	?	?	?	6,000
3	?	?	?	?	2,500
Demanda	6,000	4,000	2,000	1,500	

Figura 2.5 Representación del problema

```
min 3x11+2x12+7x13+6x14+7x21+5x22+2x23+3x24+2x31+5x32+4x33+5x34
st
x11+x12+x13+x14<5000
x21+x22+x23+x24<6000
x31+x32+x33+x34<2500
x11+x21+x31=6000
x12+x22+x32=4000
x13+x23+x33=2000
x14+x24+x34=1500
end
```

Figura 2.6 Script proporcionado a LINDO.

Información de salida:

```
LP OPTIMUM FOUND AT STEP 6
OBJECTIVE FUNCTION VALUE                39500.00

VARIABLE        VALUE          REDUCED COST
X11       3500.000000          0.000000
X12       1500.000000          0.000000
X13          0.000000          8.000000
X14          0.000000          6.000000
X21          0.000000          1.000000
X22       2500.000000          0.000000
X23       2000.000000          0.000000
X24       1500.000000          0.000000
X31       2500.000000          0.000000
X32          0.000000          4.000000
X33          0.000000          6.000000
X34          0.000000          6.000000
```

Figura 2.7 Salida de LINDO

En la Figura 2.7, se aprecia que el costo mínimo de transporte es de 39,500 y las unidades a enviar por cada ruta, para que el costo de transporte sea el mínimo satisfaciendo las restricciones, son mostradas en la Figura 2.8:

	1	**2**	**3**	**4**	**Capacidad**
1	3,500	1,500			5,000
2		2,500	2,000	1,500	6,000
3	2,500				2,500
Demanda	6,000	4,000	2,000	1,500	

Figura 2.8 Unidades que minimizan la función objetivo

2.1.2.3 Elementos asociados a la programación lineal

Existen diversos elementos asociados a la programación lineal, que permiten obtener mayor conocimiento de la naturaleza del problema analizado. Los elementos son:
Los costos reducidos, las variables de holgura, los precios sombra, el análisis de sensibilidad de los coeficientes y el análisis de sensibilidad de las restricciones.

- **Costos Reducidos (REDUCED COST)**

Las variables diferentes de cero de la solución óptima tienen un costo reducido de CERO. Por ejemplo $x_{11}, x_{12}, x_{22}, x_{23}, x_{24}, x_{31}$

Para el resto de las variables el costo reducido (Figura 2.9) representa la penalidad que implicaría pagar si se introduce la variable a la solución. Si el tipo de objetivo es minimizar, entonces se debe entender la penalidad como un incremento en el costo de la función objetivo. Si el tipo de objetivo es maximizar, entonces se debe entender la penalidad como una disminución en el costo de la función objetivo.

VARIABLE	VALUE	REDUCED COST
X11	3,500.000000	0.000000
X12	1,500.000000	0.000000
X22	2,500.000000	0.000000
X23	2,000.000000	0.000000
X24	1,500.000000	0.000000
X31	2,500.000000	0.000000

Figura 2.9 Resultados de costos reducidos del ejemplo resuelto

Por ejemplo, si consideramos la variable x_{14} de nuestro ejemplo:

VARIABLE	VALUE	REDUCED COST
X14	0.000000	6.000000

Figura 2.10 Costo reducido de la variable x_{14} del ejemplo resuelto

Costo de la función objetivo:	:	39,500
Objetivo de la función objetivo	:	Minimizar
Costo Reducido	:	6

Si se introduce la variable x_{14} a la solución, entonces por cada unidad que se incremente esta variable el costo de la función objetivo sufre un incremento equivalente al costo reducido de la variable. En este caso como el costo reducido de la variable es 6, si se hace $x_{14} = 1$, la función objetivo se incrementa de 39,500.00 a 39,506.00.

- **Variables de holgura (SLACK OR SURPLUS)**

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	3.000000
3)	0.000000	0.000000
4)	0.000000	4.000000
5)	0.000000	-6.000000
6)	0.000000	-5.000000
7)	0.000000	-2.000000
8)	0.000000	-3.000000

Figura 2.11 Relación de variables de holgura

Las variables de holgura (Figura 2.11) indican en que medida la solución óptima satisface cada restricción:

- Para restricciones <= se refieren como SLACK
- Para restricciones >= se refieren como SURPLUS
- Si la restricción es exactamente satisfecha para la igualdad, entonces el valor de SLACK/SURPLUS es CERO.

Por ejemplo para la restricción 3, referenciada como ROW 4 (Figura 2.12)

ROW	SLACK OR SURPLUS	DUAL PRICES
4)	0.000000	4.000000

Figura 2.12 Variables de holgura de la restricción 3

Restricción 3

$$x_{31} + x_{32} + x_{33} + x_{34} \leq 2,500 \quad \text{.....(3)}$$

Sustituyendo los valores obtenidos en la figura 2.9, obtenemos:

$$\begin{array}{lllll} x_{31} & +x_{32} & +x_{33} & +x_{34} \leq 2,500 & \text{.....(3)} \\ 2,500 & +0 & +0 & +0 \leq 2,500 & \text{.....(3)} \\ 2,500 & & & \leq 2,500 & \text{.....(3)} \end{array} \qquad (2.5)$$

Se observa que el valor del lado izquierdo de la restricción es igual al valor del lado derecho de la restricción, lo cual significa que las unidades del recurso disponible de la restricción 3, están agotadas.

- Precios sombra (DUAL PRICES)

ROW	DUAL PRICES
2)	3.000000
3)	0.000000
4)	4.000000
5)	-6.000000
6)	-5.000000
7)	-2.000000
8)	-3.000000

Figura 2.13 Relación de precios sombra

Los precios sombra (Figura 2.13), representan la penalidad que implicaría pagar por cada unidad adicional del recurso disponible especificado en las restricciones. El incremento o decremento del costo de la función objetivo depende del tipo de objetivo y del valor positivo o negativo de los precios sombra.

A continuación, en la Figura 2.14, se describe el comportamiento de los precios sombra:

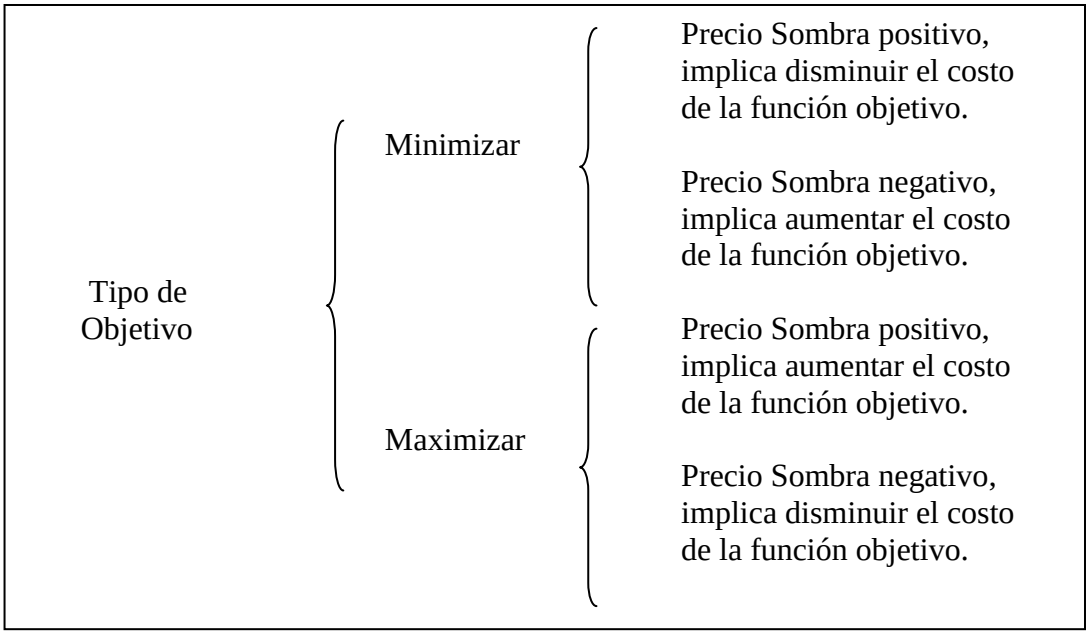

Figura 2.14 Comportamiento del uso de los precios sombra

Por ejemplo para la restricción 3 (Figura 2.15), referenciada como ROW 4

ROW	SLACK OR SURPLUS	DUAL PRICES
4)	0.000000	4.000000

Figura 2.15 Precio sombra de la restricción 3

Dado que el slack es CERO se concluye que ya no hay más recurso disponible, sin embargo si se incrementa en una unidad adicional dicho recurso (ya agotado), entonces el costo de la función objetivo disminuye en $4.

En tal caso la función objetivo disminuye de $39500.00 a $39496.00.

de

$$x_{31} + x_{32} + x_{33} + x_{34} \leq 2,500 \quad \text{.....(3)}$$

a

$$x_{31} + x_{32} + x_{33} + x_{34} \leq 2,501 \quad \text{.....(3)} \tag{2.6}$$

	Antes	Después
Costo de la función objetivo:	39,500	39,496

- **Análisis de la sensibilidad de los coeficientes de las variables de la función objetivo**

RANGES IN WHICH THE BASIS IS UNCHANGED:

	OBJ COEFFICIENT RANGES		
VARIABLE	CURRENT COEF	ALLOWABLE INCREASE	ALLOWABLE DECREASE
X11	3.000000	1.000000	4.000000
X12	2.000000	3.000000	1.000000
X13	7.000000	INFINITY	8.000000
X14	6.000000	INFINITY	6.000000
X21	7.000000	INFINITY	1.000000
X22	5.000000	1.000000	3.000000
X23	2.000000	6.000000	INFINITY
X24	3.000000	6.000000	INFINITY
X31	2.000000	4.000000	INFINITY
X32	5.000000	INFINITY	4.000000
X33	4.000000	INFINITY	6.000000
X34	5.000000	INFINITY	6.000000

Figura 2.16 Relación de análisis de sensibilidad de los coeficientes

El análisis de la sensibilidad de los coeficientes (Figura 2.16), permite saber las unidades que se pueden incrementar o decrementar los coeficientes de las variables de la función objetivo, sin causar un cambio en los valores de las variables en la solución óptima. Si consideramos la variable x_{12}

$$Min \quad \begin{aligned}&(3x_{11} + 2x_{12} + 7x_{13} + 6x_{14}) + (7x_{21} + 5x_{22} + 2x_{23} + 3x_{24}) \\ &+(2x_{31} + 5x_{32} + 4x_{33} + 5x_{34})\end{aligned} \tag{2.7}$$

```
RANGES IN WHICH THE BASIS IS UNCHANGED:

                        OBJ COEFFICIENT RANGES
VARIABLE         CURRENT        ALLOWABLE        ALLOWABLE
                  COEF          INCREASE         DECREASE
     X12        2.000000         3.000000         1.000000
```

Figura 2.17 Análisis de sensibilidad del coeficiente de la variable x_{12}

El valor actual del coeficiente es de 2 (Figura 2.17). Entonces, este coeficiente puede ser incrementado un máximo de 3 y decrementado un máximo de 1, sin que cambien los valores de la solución óptima.

Por lo tanto, el rango de modificación del coeficiente de la variable x_{12} sin afectar los valores de las variables de la solución óptima es: $1 \le C_{12} \le 5$

Comprobación:

Resolviendo el mismo problema de la figura 2.4, pero incrementando en 1 el coeficiente de la variable x_{12} en la función objetivo:

$$Min \quad \begin{array}{l}(3x_{11}+3x_{12}+7x_{13}+6x_{14})+(7x_{21}+5x_{22}+2x_{23}+3x_{24}) \\ +(2x_{31}+5x_{32}+4x_{33}+5x_{34})\end{array} \tag{2.8}$$

En la Figura 2.18, se muestran los valores de salida obtenidos en la comprobación del análisis de sensibilidad de coeficientes, variable x_{12}:

```
LP OPTIMUM FOUND AT STEP      0
       OBJECTIVE FUNCTION VALUE
       1)      41000.00

 VARIABLE        VALUE          REDUCED COST
      X11      3500.000000          0.000000
      X12      1500.000000          0.000000
      X13         0.000000          7.000000
      X14         0.000000          5.000000
      X21         0.000000          2.000000
      X22      2500.000000          0.000000
      X23      2000.000000          0.000000
      X24      1500.000000          0.000000
      X31      2500.000000          0.000000
      X32         0.000000          3.000000
      X33         0.000000          5.000000
      X34         0.000000          5.000000
```

Figura 2.18 Valores de Salida

Comparando las figuras 2.7 y 2.18, podemos comprobar que los valores (columna VALUE) de las variables (columna VARIABLE) no cambiaron.

- **Análisis de la sensibilidad de los recursos disponibles (RHS)**

	RIGHTHAND SIDE RANGES		
ROW	CURRENT RHS	ALLOWABLE INCREASE	ALLOWABLE DECREASE
2	5000.000000	2500.000000	0.000000
3	6000.000000	INFINITY	0.000000
4	2500.000000	2500.000000	0.000000
5	6000.000000	0.000000	2500.000000
6	4000.000000	0.000000	2500.000000
7	2000.000000	0.000000	2000.000000
8	1500.000000	0.000000	1500.000000

Figura 2.19 Relación de análisis de sensibilidad de los recursos disponibles

El análisis de la sensibilidad de los recursos (Figura 2.19), permite determinar el número de unidades en que se pueden incrementar o decrementar el recurso de cada restricción (RHS), sin causar un cambio en los valores de las variables de la solución óptima.

Por ejemplo, si consideramos el renglón 3 (Figura 2.20), restricción 2, Capacidad de la planta 2:

	RIGHTHAND SIDE RANGES		
ROW	CURRENT RHS	ALLOWABLE INCREASE	ALLOWABLE DECREASE
3	6000.000000	INFINITY	0.000000

Figura 2.20 Análisis de sensibilidad de la restricción 2

El valor del Current RHS es de 6000. Entonces, el lado derecho de la restricción (Current RHS), puede ser incrementado un máximo de infinito y decrementado un máximo de 0.

Por lo tanto, el rango de modificación del lado derecho de la restricción sin afectar los valores de las variables de la solución óptima es:

$$6000 < \text{Capacidad de la Planta 2} < \text{Infinito}$$

Comprobación:

Usando el mismo problema de la Figura 2.4, pero con la restricción 2 alterada de la siguiente manera:

$$\begin{aligned} &\text{de} \\ &\quad x_{21} + x_{22} + x_{23} + x_{24} \leq 6,000 \quad \text{.....(2)} \\ &\text{a} \\ &\quad x_{21} + x_{22} + x_{23} + x_{24} \leq 7,000 \quad \text{.....(2)} \end{aligned} \tag{2.9}$$

Se obtienen los siguientes valores de salida:

```
 LP OPTIMUM FOUND AT STEP      0

        OBJECTIVE FUNCTION VALUE

        1)      41000.00

 VARIABLE        VALUE          REDUCED COST
      X11      3500.000000          0.000000
      X12      1500.000000          0.000000
      X13         0.000000          7.000000
      X14         0.000000          5.000000
      X21         0.000000          2.000000
      X22      2500.000000          0.000000
      X23      2000.000000          0.000000
      X24      1500.000000          0.000000
      X31      2500.000000          0.000000
      X32         0.000000          3.000000
      X33         0.000000          5.000000
      X34         0.000000          5.000000
```

Figura 2.21 Valores de Salida

Comparando las figuras 2.7 y 2.21, podemos comprobar que los valores (columna VALUE) de las variables (columna VARIABLE) no cambiaron.

2.2 Matrices Dispersas

Una matriz dispersa, es aquella que contiene solo un pequeño número de elementos diferentes de cero. La Figura 2.22, muestra un ejemplo de una matriz dispersa [LINDO].

$$\begin{bmatrix} 3 & 0 & 0 & 2 \\ 0 & 26 & 0 & 39 \\ 4 & 5 & 8 & 0 \\ 0 & 7 & 1 & 0 \end{bmatrix}$$

Figura 2.22 Ejemplo de matriz dispersa

2.2.1 Representación de matrices dispersas mediante 3 vectores.

Debido a que al almacenar la matriz en memoria se desperdicia espacio con los elementos nulos, ésta se representa usando tres (u opcionalmente cuatro) vectores, de tal forma que los elementos nulos no se almacenan. Dado que la mayoría de los coeficientes de la matriz en modelos de programación matemática del mundo real son cero o nulos, este esquema de almacenamiento es muy adecuado para la representación en memoria de dichos modelos [LINDO].

2.2.2 Procedimiento de representación de matrices dispersas mediante 3 vectores

Primero se crea un vector denominado vector de Valores, el cual contiene todos los elementos distintos de cero de la matriz, ordenados por la columna. Para la matriz, del ejemplo anterior, este vector sería: [3 4 26 5 7 8 1 2 39].

Los elementos 3 y 4 de la primera columna aparecen primero, cuyos índices son el 0 y el 1 respectivamente. Después aparecen los elementos 26, 5 y 7 de la segunda columna, cuyos índices son el 2, 3 y 4 respectivamente. Aparecen después los elementos 8 y 1 de la tercera columna, cuyos índices son el 5 y 6 respectivamente. Finalmente aparecen los elementos de la cuarta columna 2 y 39, cuyos índices son el 7 y 8 respectivamente. Como se observa en la construcción del vector de Valores todos los ceros de la matriz dispersa son descartados.

Ahora se crea un vector denominado Inicio-Columna, en el cual se registra qué índices del vector de Valores representan el comienzo de una columna de la matriz original.

Para el vector de Valores anterior el vector Inicio–Columna es: [0 2 5 7 9].

Este vector contiene un elemento adicional, el cual indica en que índice del vector de Valores termina la última columna. Éste último elemento es igual a la longitud del vector de Valores. El vector Inicio-Columna permite determinar a que columna pertenece cada elemento del vector de Valores.

Una vez realizado el proceso anterior, la única información adicional que se requiere para especificar completamente la matriz dispersa, es el número de fila de cada elemento del vector de Valores. Esta información se almacena en un tercer vector, el vector Índice-Fila. Este vector es de la misma longitud que el vector de Valores y cada uno de sus elementos indica la fila a la que pertenece el elemento correspondiente del vector Valores. Por ejemplo, el primer elemento del vector de Valores, el número 3, pertenece a la fila 0, así que el primer elemento del vector Índice-Fila es 0. De igual forma, el segundo elemento del vector de Valores es 4 y pertenece a la fila 2, así que el segundo elemento del vector Índice-Fila es 2. De manera similar se determinan todos los elementos del vector Índice - Fila [0 2 1 2 3 2 3 0 1].

2.3 Interfase de programación de LINDO 2.0

La interfase de programación de LINDO 2.0 es un conjunto de funciones que pueden ser invocadas desde diversos lenguajes de programación para resolver problemas de programación lineal. Para demostrar como se resuelve un problema de programación lineal usando esta interfase, se utilizará el ejemplo de la Figura 2.23.

$$
\begin{array}{llrrrcr}
max & z = & 10x_{11} & +150x_{12} & +30x_{13} & & \\
st & & & & & & \\
 & & x_{11} & +3x_{12} & & \leq & 100 \\
 & & 10x_{11} & +10x_{12} & +5x_{13} & \geq & 1000 \\
 & & x_{11} & & & \leq & 200 \\
 & & & x_{12} & & \leq & 20 \\
 & & & & x_{13} & \leq & 50
\end{array}
$$

Figura 2.23 Ejemplo de un problema de programación lineal

2.3.1 Representación en la matriz de restricciones

El primer paso para resolver un problema de programación lineal es representar la matriz de restricciones utilizando la representación de tres vectores. En la Figura 2.24, indica la matriz de restricciones del problema considerado:

$$\begin{bmatrix} 1 & 3 & 0 \\ 10 & 10 & 5 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figura 2.24 Matriz de restricciones del problema considerado

Entonces su representación mediante tres vectores es la siguiente:

vector de Valores: [1 10 1 3 10 1 5 1]

vector Inicio-Columna: [0 3 6 8]

vector Índice -Fila: [0 1 2 0 1 3 1 4]

2.3.2 Estructuras de datos utilizadas por el API de LINDO

Una vez determinada la representación mediante tres vectores de la matriz de restricciones, todos los componentes del problema se definen a través de las estructuras de datos que se describen en la Tabla 2.1.

La primera columna de la tabla, contiene el nombre de la estructura requerida por LINDO; la segunda, el tipo de dato; la tercera, la descripción del componente del problema que se registra en la estructura y la última, muestra el contenido correspondiente a los componentes del ejemplo considerado.

Tabla 2.1 Estructuras de datos utilizadas por la interfaz de LINDO

Dato	Tipo	Descripción	Ejemplo
pachContypes	Char *	Tipo de restricción	L, G, L, L
nCons	Int	Número de restricciones	4
padC	double*	Coeficientes de la función objetivo	10, 150,30
nAnnz	Int	Número de no ceros en la matriz de restricciones	8
padAcoef	double*	Coeficientes de los no ceros en la matriz de restricción	1, 10, 1, 3, 10, 1, 5, 1
paiArows	Int *	Índice de fila de los no ceros	0, 1, 2, 0, 1, 3, 1, 4
nVars	Int	Número de variables	3
paiAcols	Int *	Índice del primer no ceros en cada columna	0, 3, 6
padB	double*	Coeficientes del lado derecho de las restricciones	100, 1000, 200, 20, 50
padL	double*	Límites bajos de cada variable	Null
padU	double*	Límites altos de cada variable	Null
dObjconst	double	Valor constante agregado a la función objetivo	0
pacAcols	Int *	Longitud de cada columna	null
tipo_var	Int *	Tipo de variable, 1 si es entera, 0 si no lo es	1

2.3.3 Codificación en lenguaje C.

Para resolver el problema de programación lineal, utilizando el lenguaje de programación C, se deben realizar los siguientes pasos:

1. Crear un entorno de LINDO.
2. Crear un modelo en el entorno.
3. Definir el modelo usando las estructuras descritas.
4. Ejecutar la optimización.
5. Recuperar los resultados de la optimización.
6. Eliminar el entorno de LINDO.

La Figura 2.25 muestra el código fuente correspondiente al ejemplo considerado. Mayores detalles relativos a este proceso se pueden encontrar en [LINDO].

```
#include <stdlib.h>   #include <stdio.h>
#include "lindo.h"    #include "license.h"
#define APIERRORSETUP  \
   int nErrorCode; \
   char cErrorMessage[LS_MAX_ERROR_MESSAGE_LENGTH] \
#define APIERRORCHECK  \
   if (nErrorCode) \
   { \
      if ( pEnv) \
      { \
         LSgetErrorMessage( pEnv, nErrorCode, \
          cErrorMessage); \
         printf("Código de error=%d:  %s\n", nErrorCode, \
          cErrorMessage); \
      } else {\
         printf( "Error fatal\n"); \
      } \
      exit(1); \
   } \
int main() {
   APIERRORSETUP;
/* Número de restricciones */      int nM = 4;
/* Número de variables */       int nN = 3;
/* tipo entorno de LINDO */      pLSenv pEnv;
/* tipo modelo de LINDO*/   pLSmodel pModel; int nSolStatus;
```

Figura 2.25 Código fuente del problema considerado

```
/* 1. Crear un entorno de LINDO */

   pEnv = LScreateEnv ( &nErrorCode, MY_LICENSE_KEY);
   if ( nErrorCode == LSERR_NO_VALID_LICENSE)
   {
      printf( "Licencia invalida!\n");
      exit( 1);
   }
   APIERRORCHECK;

/* 2. Crear un modelo en el entorno*/

   pModel = LScreateModel ( pEnv, &nErrorCode);
   APIERRORCHECK;
   {

/* 3. Definir el modelo */

/* Dirección de optimización*/
      int nDir = LS_MAX;
/* Término constante en la función objetivo*/
      double dObjConst = 0.;
/* Coeficientes de la función objetivo */
      double adC[2] = { 10., 150., 30.};
/* Lados derechos de las restricciones*/
      double adB[3] = { 100., 1000., 200., 20.};
/* Tipos de restricciones*/
      char acConTypes[3] = {'L', 'G', 'L', 'L'};
/* Número de elementos no cero en la matriz de
restricciones*/
      int nNZ = 8;
/* Vector Valores de la matriz de restricciones */
      double adA[4] = { 1., 10., 1., 3., 10., 1., 5., 1.};
/* Vector Inicio-Columna de la matriz de restricciones*/
      int anBegCol[3] = { 0, 3, 6, nNZ};
/* Vector Índice-Fila de la matriz de restricciones*/
      int anRowX[4] = { 0, 1, 2, 0, 1, 3, 1, 4};
/* Longitud de cada columna (se debe especificar cuando la
representación de la matriz de restricciones incluya
elementos que son cero, en otro caso se le asigna NULL)*/
      int *pnLenCol = NULL;
```

Figura 2.25 Continuación …

```
/* Limites inferiores y superiores de las variables de
decisión del   problema. Los valores por omisión son cero e
infinito. */
double *pdLower = NULL, *pdUpper = NULL;
/* Cargar el modelo a memoria utilizando la función
LSloadLPData */
      nErrorCode = LSloadLPData( pModel, nM, nN, nDir,
       dObjConst, adC, adB, acConTypes, nNZ, anBegCol,
       pnLenCol, adA, anRowX, pdLower, pdUpper);
      APIERRORCHECK;
   }
 /* 4. Ejecutar la optimización */
   nErrorCode = LSoptimize( pModel,
    LS_METHOD_PSIMPLEX, &nSolStatus);
   APIERRORCHECK;
   if (nSolStatus == LS_STATUS_OPTIMAL ||
       nSolStatus == LS_STATUS_BASIC_OPTIMAL)
   {
 /* 5. Recuperar los resultados de la optimización */
      int i;
      double adX[ 2], dObj;
 /* Obtener el valor óptimo de la función objetivo*/
      nErrorCode = LSgetInfo( pModel, LS_DINFO_POBJ, &dObj)
;
      APIERRORCHECK;
      printf( "Valor objetivo óptimo= %g\n", dObj);
 /* Obtener los valores de las variables de decisión del
problema*/
      nErrorCode = LSgetPrimalSolution ( pModel, adX);
      APIERRORCHECK;
      printf ("Primal values \n");
      for (i = 0; i < nN; i++)
printf( " x[%d] = %g\n", i,adX[i]); printf ("\n");
   }
   else   {
     printf( "Solución óptima no alcanzada -- código: %d\n",
     nSolStatus);
   }
 /* 6. Eliminar el entorno de LINDO */
   nErrorCode = LSdeleteEnv( &pEnv);
   printf("Presione <Enter> ...");
   getchar();
}
```

Figura 2.25 Continuación …

2.4 Trabajos relacionados

Dado que el problema de abastecimiento internacional esta muy relacionado con el problema de ubicación de plantas, a continuación se describen algunos de los trabajos en los que se aborda el problema de la ubicación de plantas:

Kraup y Pruzan, hicieron una investigación del problema de ubicación de plantas, considerando las siguientes versiones: no capacitada, capacitada, dinámica (multiperiodo) y estocástica. Una versión estocástica, es mencionada muy ligeramente y no reportan resultados [Kraup 1979].

Verter y Dincer, analizan el problema ubicación de plantas y consideran las siguientes versiones del problema: no capacitada, capacitada, estocástica y el problema de la ubicación internacional de las plantas [Verter 1992].

Los siguientes trabajos abordan la versión estocástica del problema de la ubicación internacional de las plantas.

Louveaux y Peters, presentaron un problema basado en escenarios en el cual la capacidad de las plantas se determina como resultado de una decisión inicial. A pesar de la limitación de la capacidad de las plantas, los autores lo refieren como un problema de capacidad infinita [Louveaux 1992].

Jucker y Carlson, analizan el problema considerando un solo producto, un solo periodo y que el precio y la demanda son inciertas. Presentan dos tipos diferentes de plantas: las que son capaces de controlar el precio y la demanda de sus productos y las que no tienen control de estos factores. La solución del problema de localización de las plantas requiere la solución de un modelo de programación cuadrática entera. Para simplificar la solución de este problema se introduce el concepto de planta dominante, donde se considera que cada uno de los mercados solo puede ser atendido por una sola planta. Este supuesto permite transformar la parte cuadrática del problema en un subproblema lineal [Jucker 1976].

Hodder y Jucker, presentan un modelo de un único periodo, un sólo producto. Modelan un control de precios y una demanda que depende del precio y resuelven el problema utilizando plantas dominantes [Hodder 1982].

Hodder y Dincer, conciben simultáneamente el problema de la ubicación de las plantas y las decisiones de financiamiento. La rentabilidad de una planta depende de la ubicación de las plantas. Formulan un modelo de un solo producto y un solo periodo. Suponen que toda la producción de las plantas, hasta un límite determinado, se podrá vender con una ganancia unitaria incierta [Hodder 1986].

Gutiérrez y Kouvelis, refieren el problema de la generación de escenarios para modelar precios inciertos y resolver el problema simple de ubicación de plantas no capacitado. Esto es equivalente a un modelo del problema del abastecimiento internacional con un criterio regret mínimax. Sin embargo, sus resultados deben ser examinados detalladamente, ya que las soluciones reportadas pueden ser peores que el valor esperado de las soluciones correspondientes a cada uno de los escenarios, cuando se utiliza un conjunto grande de escenarios [Gutiérrez 1995].

Lawrence y Buss, consideran la posibilidad de producir en diferentes países para obtener ventaja de las variaciones de la tasa de cambio de las monedas locales. Modelan la producción de un único producto en dos países diferentes, cada uno de los cuales tiene una demanda determinada [Lawrence 1999].

La Tabla 2.2, muestra un resumen de las principales características de los trabajos anteriores.

Tabla 2.2 Resumen de artículos que tratan el problema de la ubicación de la planta.

Autores	Robusto	Periodo	Producto	Capacidad finita	Estocástico
Louveraux y Peters, 1992	escenarios	único	único	sí	sí
Jucker y Carlson, 1976		sí	sí	sí	sí
Hodder y Jucker, 1982,1985a, 1985b		sí	sí		
Haug, 1985		Múltiple	Único		
Hodder y Dincer, 1986		Único	Único		
Cohen y Lee, 1989		Único	Múltiple		
Gutierrez y Kouvelis, 1995				Infinita	

Algunos de los trabajos en que se aborda el problema del abastecimiento internacional con incertidumbre son los siguientes:

Kouvelis y Yu, tratan sobre el problema de abastecimiento internacional con capacidad infinita en los proveedores. Aplican un enfoque robusto en la formulación del problema y utilizan una función objetivo de tipo minimax. El método de

solución que proponen consiste en encontrar la solución para el peor de los escenarios posibles. La principal desventaja de este enfoque es que la solución que se produce es demasiado pesimista y en general se pudiera encontrar una mejor solución en base a un escenario promedio [Kouvelis y Yu 1997].

Escudero, presentó un trabajo relacionado al abastecimiento en los procesos de manufactura. Donde el costo esperado es el único término a optimizar en la función objetivo [Escudero 1993].

Del problema robusto del abastecimiento internacional con capacidad finita, se conocen los siguientes trabajos:

La formulación robusta del problema ROCIS fue propuesta por primera vez por González- Velarde y Laguna en [González 2004]. González Velarde y Laguna proponen un método de solución que incorpora mecanismos de búsqueda tabú. El proceso consiste en construir una solución inicial seguida de una búsqueda voraz en la vecindad de dicha solución. Como la elección de la solución inicial determina la eficiencia del proceso, dicha solución se construye dando preferencia a los proveedores de mayor capacidad y con un menor costo fijo.

Por otra parte, González-Velarde y Martí en [González 2006], proponen un método de solución basado en reencadenamiento de trayectorias. El método, incorpora el costo de envío de cada proveedor a todas las plantas, como parte de la estrategia de prioridades que se utiliza para construir un conjunto de soluciones iniciales. Los propios autores señalan, que esta manera de incorporar el costo de envío parece ser demasiado pesimista. Los resultados presentados en el artículo, no permiten determinar en que medida la mejora reportada depende del cambio en la estrategia de prioridades para la generación de soluciones iniciales.

La Tabla 2.3, muestra las estrategias de selección de proveedores, utilizadas en los métodos de solución del problema de abastecimiento internacional con capacidad finita reportados en la literatura.

Tabla 2.3 Características relevantes

Artículo	**Método de solución**	**Estrategia**
Laguna y Velarde, 2004	Búsqueda Tabú.	$G_i = \frac{f_i}{b_i}$
Marti y Velarde, 2006	Re-encadenamiento de trayectorias	$G_i = \frac{f_i + \left(\sum_{s \in S} p_s \left(\sum_{j=1}^{n} c_{ij} e_{is} \right) \right)}{b_i}, \forall j \in N, i \in M$

La estrategia de selección de proveedores permite la construcción de soluciones iniciales de calidad, lo cual determina el desempeño del método de solución. Una de las cuestiones actualmente abiertas en el estudio de este problema es la determinación de nuevas estrategias de prioridades que permitan mejorar el desempeño de dicha solución [González 2006].

En esta tesis, se proponen diferentes estrategias de prioridades para la construcción de soluciones iniciales y se presenta evidencia experimental del impacto en el desempeño de la solución tabú del problema ROCIS.