

Introducción a GLPK en *Optimización de Problemas de Producción*

Docentes: Fernando Islas - Carlos Testuri

Colaboradores: Héctor Cancela - Antonio Mauttone - Pedro Piñeyro

Depto. Investigación Operativa. Instituto de Computación.
Facultad de Ingeniería. UdelaR

2023

Sistema informático para la resolución de problemas de programación lineal (LP) y programación lineal entera mixta (MIP), integrado por

- Lenguaje de modelado algebraico (GNU MathProg modeling language)
- Solver que utiliza diferentes algoritmos: métodos simplex primal y dual, punto interior primal-dual y branch and cut.

- Los problemas se escriben en un archivo de texto plano utilizando los elementos del lenguaje
- **Descripción del modelo:** Formado por una secuencia de **sentencias**
- **Datos del modelo:** Formado por una secuencia de **bloques de datos**

Descripción del modelo (Sentencias)

- Las sentencias son la la unidad básica del lenguaje
- Toda sentencia finaliza con un punto y coma (;)
- **Sentencias Declarativas:** Sirven para definir **objetos del modelo**
- **Sentencias Funcionales:** Para realizar acciones específicas
(No las veremos en el curso)

Descripción del modelo (Objetos)

- Es definido mediante un nombre simbólico que lo identifica
- Los objetos pueden ser: escalares o n -dimensionados ($n \geq 1$)
- **Objetos Escalares:** Los referenciamos por medio del nombre; por ejemplo t
- **Objetos Vectoriales**(1-dimensión): Los referenciamos por medio del nombre y un subíndice entre corchetes; por ejemplo: $nutriente[i]$
- **Objetos n -Dimensionados:** Los referenciamos por medio del nombre y n subíndices entre corchetes; por ejemplo para $n=2$ (matriz), $contenido[i,j]$

Descripción del modelo (Conjuntos, Parámetros y Variables)

- Los **conjuntos** definen el dominio de los índices utilizados en el modelo
- Los **parámetros** representan datos de la realidad
- Las **variables** representan magnitudes que están bajo el control de quién toma las decisiones

Definición y Sintáxis	Ejemplo
Conjuntos <code>set nombre;</code>	<code>set A;</code>
Parámetros <code>param nombre dominio;</code>	<code>param precio{i in A};</code>
Variables de Decisión <code>var nombre dominio expresión;</code>	<code>var x{i in A} >= 0;</code>

Descripción del modelo (Restricciones)

- Las **restricciones** representan exigencias que debe cumplir la solución que buscamos

Definición y Sintaxis

Restricciones

s.t. *nombre dominio : expresión = expresión;*

s.t. *nombre dominio : expresión ≤ expresión;*

s.t. *nombre dominio : expresión ≥ expresión;*

Ejemplo

s.t. *demanda{j in N} :*

*sum{i in A} x[i] * contenido[i,j] ≥ requisito[j];*

- La **función objetivo** representa la expresión que queremos optimizar

Definición y Sintáxis	Ejemplo
Función Objetivo maximize <i>nombre dominio</i> : <i>expresión</i> ;	minimize <i>costo</i> : <i>sum{i in A} precio[i] * x[i]</i> ;

- Es una construcción muy útil al momento de documentar nuestros modelos
- Sirven para hacer anotaciones legibles
- Estas anotaciones son ignoradas por el intérprete que procesa el modelo
- **Comentarios de línea:** El carácter numeral (#) comenta todo lo que se encuentre a su derecha hasta el final de la línea
- **Comentarios de bloque:** La secuencia barra asterisco (/*) comenta todos los caracteres que se encuentren hasta la secuencia asterisco barra (*/)

- La primera forma de proporcionarle datos al modelo es por medio del **operador de asignación :=** (dos puntos igual) en la declaración (definición) del objeto. Por ejemplo:

set A := Leche Carne Arroz Papa Tomate;

- La segunda es por medio de bloques de datos en la **sección de datos del modelo**

Datos del modelo (Sección de datos)

- La sección de datos del modelo puede estar definida en el mismo archivo que la descripción del modelo, luego de la sentencia **data**;
- En esta sección las expresiones no están permitidas
- Por medio de **bloques de datos** se asignan valores a los *Conjuntos* y a los *Parámetros*

```
sentencia;  
    . . .  
sentencia;  
data;  
bloque de datos;  
    . . .  
bloque de datos;  
end;
```

Datos del modelo (Sección de datos) (2)

- Aunque por lo general vamos a querer tener los datos del modelo en un archivo diferente, de modo de poder instanciar un mismo modelo con distintos juegos de datos

```
sentencia;  
sentencia;  
  . . .  
sentencia;  
end;
```

Archivo del modelo

```
data;  
bloque de datos;  
bloque de datos;  
  . . .  
bloque de datos;  
end;
```

Archivo de datos

Datos del modelo (Bloques de datos)

Bloque de datos	Ejemplo
Conjuntos set <i>nombre</i> := t_1, \dots, t_n ;	set <i>A</i> := <i>Leche Carne Arroz Papa Tomate</i> ;
Parámetros (Escalar) param <i>nombre</i> := v ;	param <i>T</i> := 4;
Parámetros (Vector) param <i>nombre</i> := $t_1, v_1,$ $t_2, v_2,$ \dots $t_{n-1}, v_{n-1},$ t_n, v_n ;	param <i>precio</i> := Leche 24 Carne 190 Arroz 32 Papa 25 Tomate 50;

Datos del modelo (Bloques de datos)

Bloque de datos

Parámetros (Matriz)

param nombre	:	c_1	c_2	...	c_n	:=
	r_1	a_{11}	a_{12}	...	a_{1n}	
	r_2	a_{21}	a_{22}	...	a_{2n}	
		
	r_m	a_{m1}	a_{m2}	...	a_{mn}	;

Ejemplo

Parámetros (Matriz)

param contenido	:	Gr	Pr	Ca	:=
	<i>Leche</i>	20	34	49	
	<i>Carne</i>	250	180	0	
	<i>Arroz</i>	2,5	67	780	
	<i>Papa</i>	1	21	170	
	<i>Tomate</i>	2	11	47;	

Problema de la dieta

El problema trata de la selección de alimentos que satisfacen requisitos nutricionales a costo mínimo. Dado un conjunto de alimentos, con su información nutricional y precio, el objetivo es seleccionar las cantidades de alimentos a adquirir de forma de minimizar el costo de la dieta, mientras se cumple con requisitos nutricionales. Estos requisitos se establecen como cotas mínimas de algunos componentes nutricionales.

Cuadro: Contenido de nutrientes y precios por alimento

Alimento	Grasas(g)	Proteínas(g)	Carbohidratos(g)	Precio(\$)
Leche (L)	20	34	49	24
Carne (Kg)	250	180	0	190
Arroz (Kg)	2.5	67	780	32
Papa (Kg)	1	21	170	25
Tomate (Kg)	2	11	47	50
Requisitos (/día)	60	120	280	

Problema de la dieta (Formulación)

El objetivo es determinar la cantidad diaria de cada alimento a adquirir mediante las variables x_L, x_C, x_A, x_P, x_T .

Para cada nutriente existe una restricción de cantidad mínima.

$$\left\{ \begin{array}{ll} \min & 24x_L + 190x_C + 32x_A + 25x_P + 50x_T \\ \text{s.a.} & 20x_L + 250x_C + 2,5x_A + x_P + 2x_T \geq 60, \quad (\text{Grasas}) \\ & 34x_L + 180x_C + 67x_A + 21x_P + 11x_T \geq 120, \quad (\text{Proteinas}) \\ & 49x_L + 780x_A + 170x_P + 47x_T \geq 280, \quad (\text{Carbohidratos}) \\ & x_L, x_C, x_A, x_P, x_T \geq 0. \end{array} \right.$$

Implementación del Problema de la dieta (Conjuntos y Parámetros)

```
1 # PROBLEMA DE LA DIETA
2 #
3
4 set A;
5 /* Alimentos */
6
7 set N;
8 /* Nutrientes */
```

```
10 param precio {i in A};
11 /* Precio por alimento */
12
13 param requisito {j in N};
14 /* Requisitos nutricionales diarios */
15
16 param contenido {i in A, j in N};
17 /* Contenido nutricional j por alimento i */
```

Implementación del Problema de la dieta (Variables, restricciones y objetivo)

```
19 var x{i in A} >= 0;  
20 /* Cantidades a ser adquiridas */
```

```
22 minimize cost: sum{i in A} precio[i] * x[i];  
23 /* Costo minimo */
```

```
25 s.t. demanda{j in N}: sum{i in A} x[i] * contenido[i,j] >=  
    requisito[j];  
26 /* Satisfaga los requisitos diarios para el nutriente i */
```

Implementación del Problema de la dieta (Bloques de datos)

```
28 data ;
29
30 set A := Leche Carne Arroz Papa Tomate;
31
32 set N := Grasas Proteinas Carbohidratos;
33
34 param precio := Leche      24 /*($ por litro)*/
35                  Carne     190 /*($ por Kg)*/
36                  Arroz     32 /*($ por Kg)*/
37                  Papa      25 /*($ por Kg)*/
38                  Tomate    50; /*($ por Kg)*/
39
40 param requisito := Grasas      60 /*(g diarios)*/
41                  Proteinas    120 /*(g diarios)*/
42                  Carbohidratos 280; /*(g diarios)*/
```

Implementación del Problema de la dieta (Bloques de datos) (2)

```
44 param contenido :      Grasas      Proteinas  Carbohidratos :=  
45 #                      (g)          (g)          (g)  
46           Leche      20           34           49  
47           Carne     250          180           0  
48           Arroz     2.5           67          780  
49           Papa       1            21          170  
50           Tomate     2            11          47;  
51 end ;
```

El programa `glpsol` es ejecutado desde la línea de comandos. Para obtener la consola de línea de comandos en MS-Windows hay que ejecutar el programa `cmd.exe`.

En `glpsol`, para indicarle al intérprete que el archivo de entrada corresponde a una descripción de un modelo, se utiliza la opción `model`. Por medio de la opción `output` se le indica el archivo en donde se almacenará la salida con los resultados. Las opciones son anteceditas por dos guiones o signo de menos.

```
C:\> glpsol --model dieta.mod --output dieta.sol
```

Interpretación de los resultados (Información sobre el problema)

```
Problem:    dieta
Rows:      4
Columns:   5
Non-zeros: 19
Status:    OPTIMAL
Objective: cost = 80.3187251 (MINimum)
```

El solver ha encontrado un valor óptimo (mínimo en este caso) al problema *dieta* y vale \$ 80.3187251.

Interpretación de los resultados

(Información sobre el estado de la función objetivo y las restricciones)

No.	Row name	St	Activity	Lower bound	Upper bound	Marginal
1	cost	B	80.3187			
2	demanda[Grasas]	NL	60	60		0.414343
3	demanda[Proteinas]	NL	120	120		0.462151
4	demanda[Carbohidratos]	B	368.988	280		

Analizando la tabla vemos que el estado (columna St) de los requisitos nutricionales en Grasas esta acotado inferiormente (NL). Su valor marginal (valor dual o precio sombra) vale \$ 0.4143 por lo que si la restricción fuera relajada, la función objetivo mejoraría ese valor. Con los requisitos de Proteínas sucede algo similar. Sin embargo los requisitos nutricionales en Carbohidratos no están acotados (St es B) por lo que la restricción esta inactiva. Relajarla no mejorará el valor objetivo; su valor dual es cero.

Interpretación de los resultados (Información de la solución)

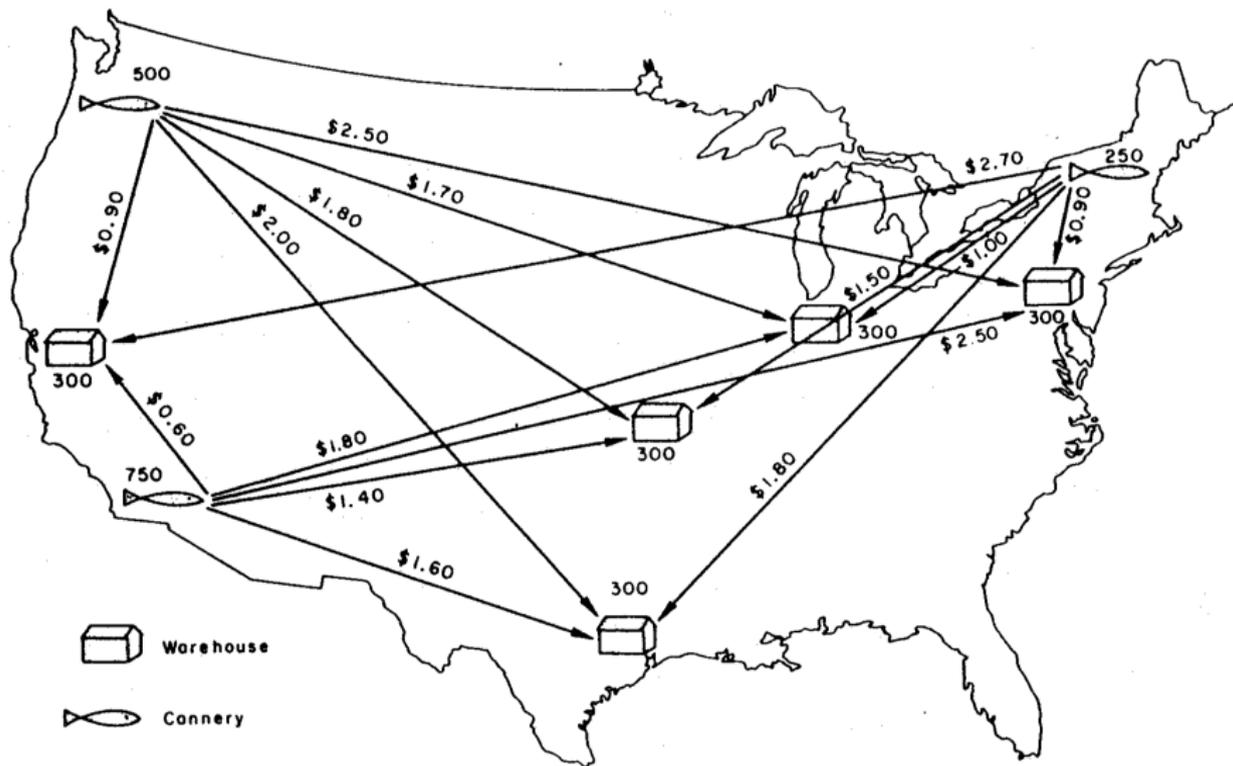
No.	Column name	St	Activity	Lower bound	Upper bound	Marginal
1	x[Leche]	B	2.96414	0		
2	x[Carne]	NL	0	0		3.22709
3	x[Arroz]	B	0.286853	0		
4	x[Papa]	NL	0	0		14.8805
5	x[Tomate]	NL	0	0		44.0876

Vemos en la columna `Activity` la solución óptima del problema. Las que presentan estado `B` son las variables básicas. La solución es entonces Leche = 2.96414 L y Arroz 0.286853 kg. Deberíamos adquirir esas cantidades diariamente para minimizar los costos y cumplir con los requisitos nutricionales. La columna `Marginal` indica los costos reducidos de las variables no básicas, ante lo que se puede confirmar que se ha alcanzado el mínimo.

Problema de distribución

El objetivo es encontrar un programa de envíos de menor costo que satisfaga los requerimientos del mercado y los suministros de las plantas. Supongamos que contamos con tres plantas de productos enlatados ubicados en Portland (Maine), Seattle y San Diego. Las plantas pueden llenar 250, 500 y 750 latas de conserva por día, respectivamente. El distribuidor opera con cinco almacenes ubicados en New York, Chicago, Topeka (Kansas City), Dallas y San Francisco. Cada uno de los almacenes puede vender 300 latas diariamente. El distribuidor desea determinar el número de latas de conserva a ser entregadas de las tres plantas a los cinco almacenes de modo que cada almacén debería obtener tantas latas como pueda vender diariamente a costo de transporte mínimo.

Problema de distribución (2)



Problema de distribución (3)

Para simplificar el modelo vamos a suponer que contamos con dos plantas (Seattle y San Diego) y tres almacenes (New York, Chicago y Topeka). En el cuadro mostramos los costos de envío (en dólares americanos) por unidad desde las plantas a los almacenes.

Cuadro: Costos de distribución (dólares por lata)

Plantas	Almacenes		
	New York	Chicago	Topeka
Seattle	2.5	1.7	1.8
San Diego	2.5	1.8	1.4

Problema de distribución (Decisiones y objetivo)

- Decisiones:

$x_{Seattle,NewYork}$

$x_{Seattle,Chicago}$

$x_{Seattle,Topeka}$

$x_{SanDiego,NewYork}$

$x_{SanDiego,Chicago}$

$x_{SanDiego,Topeka}$

Cantidad de latas de Seattle a New York

Cantidad de latas de Seattle a Chicago

Cantidad de latas de Seattle a Topeka

Cantidad de latas de San Diego a New York

Cantidad de latas de San Diego a Chicago

Cantidad de latas de San Diego a Topeka

- Objetivo:

$$\min \sum_{i \in \{Seattle, SanDiego\}} \sum_{j \in \{NewYork, Chicago, Topeka\}} \text{transporte}[i, j] \times x[i, j]$$

Problema de distribución (Restricciones)

- Para cada planta, la cantidad de conservas distribuidas debe ser menor o igual a la cantidad producida.

$$x_{Seattle,NewYork} + x_{Seattle,Chicago} + x_{Seattle,Topeka} \leq 500$$

$$x_{SanDiego,NewYork} + x_{SanDiego,Chicago} + x_{SanDiego,Topeka} \leq 750$$

- Debemos satisfacer la demanda de los almacenes.

$$x_{Seattle,NewYork} + x_{SanDiego,NewYork} \geq 300$$

$$x_{Seattle,Chicago} + x_{SanDiego,Chicago} \geq 300$$

$$x_{Seattle,Topeka} + x_{SanDiego,Topeka} \geq 300$$

- No negatividad de las variables de decisión:

$$x_{Seattle,NewYork} \geq 0, x_{Seattle,Chicago} \geq 0, x_{Seattle,Topeka} \geq 0,$$

$$x_{SanDiego,NewYork} \geq 0, x_{SanDiego,Chicago} \geq 0, x_{SanDiego,Topeka} \geq 0$$

Problema de distribución (Formulación)

$$\left\{ \begin{array}{l} \min \quad \sum_{i \in \{Seattle, SanDiego\}} \sum_{j \in \{NewYork, Chicago, Topeka\}} \text{transporte}[i,j] \times x[i,j] \\ \text{s.a.} \quad \sum_{j \in \{NewYork, Chicago, Topeka\}} x_{Seattle,j} \leq 500 \\ \quad \quad \sum_{j \in \{NewYork, Chicago, Topeka\}} x_{SanDiego,j} \leq 750 \\ \quad \quad \sum_{i \in \{Seattle, SanDiego\}} x_{i,NewYork} \geq 300 \\ \quad \quad \sum_{i \in \{Seattle, SanDiego\}} x_{i,Chicago} \geq 300 \\ \quad \quad \sum_{i \in \{Seattle, SanDiego\}} x_{i,Topeka} \geq 300 \\ \quad \quad x_{i,j} \geq 0 \text{ con } i \in \{Seattle, SanDiego\} \text{ y } j \in \{NewYork, Chicago, Topeka\} \end{array} \right.$$

Implementación del Problema de distribución (Conjuntos y Parámetros)

```
1 # PROBLEMA DE TRANSPORTE
```

```
2 #
```

```
3  
4 set P;
```

```
5 /* Plantas */
```

```
6  
7 set A;
```

```
8 /* Almacenes */
```

```
10 param capacidad{i in P};
```

```
11 /* Capacidad de la planta i */
```

```
12  
13 param demand{j in A};
```

```
14 /* Demanda del mercado j */
```

```
15  
16 param transporte{i in P, j in A};
```

```
17 /* Costo de transporte por lata de conserva */
```

Implementación del Problema de distribución (Variables, restricciones y objetivo)

```
19 var x{i in P, j in A} >= 0;  
20 /* Cantidades a ser distribuidas */
```

```
22 minimize cost: sum{i in P, j in A} transporte[i,j] * x[i,j];  
23 /* Costos totales de transporte */
```

```
25 s.t. supply{i in P}: sum{j in A} x[i,j] <= capacidad[i];  
26 /* Sujeto a la capacidad de la planta i */  
27  
28 s.t. demand{j in A}: sum{i in P} x[i,j] >= demanda[j];  
29 /* Satisfaga la demanda del mercado j */
```

Implementación del Problema de distribución (Bloques de datos)

```
31 data ;
32
33 set P := Seattle San-Diego ;
34
35 set A := New-York Chicago Topeka ;
36
37 param capacidad := Seattle      500
38                  San-Diego     750 ;
39
40 param demand := New-York      300
41                  Chicago      300
42                  Topeka       300 ;
43
44 param transporte :      New-York   Chicago   Topeka :=
45                  Seattle  2.5       1.7       1.8
46                  San-Diego 2.5       1.8       1.4 ;
47
48 end ;
```

Interpretación de los resultados

(Información sobre el estado de la función objetivo y las restricciones)

No.	Row name	St	Activity	Lower bound	Upper bound	Marginal
1	cost	B	1680			
2	supply[Seattle]	NU	500		500	< eps
3	supply[San-Diego]	B	400		750	
4	demand[New-York]	NL	300	300		2.5
5	demand[Chicago]	NL	300	300		1.7
6	demand[Topeka]	NL	300	300		1.4

Si bien la restricción *supply[Seattle]* acota la función objetivo superiormente, el valor marginal está muy próximo a 0 (ϵ , con $\epsilon \rightarrow 0$).

Por otro lado, las restricciones de mercado nos indican que si pudiéramos elegir un mercado para reducir su demanda (relajar la restricción), nos convendría elegir el de New York, dado que cada unidad decrementaría los costos 2.5 puntos en el valor óptimo.

Interpretación de los resultados (Información de la solución)

No.	Column name	St	Activity	Lower bound	Upper bound	Marginal
1	x[Seattle,New-York]	B	200	0		
2	x[Seattle,Chicago]	B	300	0		
3	x[Seattle,Topeka]	NL	0	0		0.4
4	x[San-Diego,New-York]	B	100	0		
5	x[San-Diego,Chicago]	NL	0	0		0.1
6	x[San-Diego,Topeka]	B	300	0		

Analizando la segunda tabla vemos que el programa de envío de menor costo es aquel que envía desde Seattle 200 latas a New York, y 300 a Chicago; y desde San Diego 100 a New York y 300 a Topeka cumpliendo los límites de las plantas y las demandas de los mercados. A su vez indica que enviar latas desde Seattle a Topeka o de San Diego a Chicago aumenta el costo por lata en dólares en 0.4 y 0.1 unidades respectivamente.

Problema de determinación de lotes no capacitada

El problema de determinación de lotes no capacitada se describe para un horizonte de n períodos donde, para cada período, $t \in \{1, \dots, n\}$, se debe determinar el nivel de producción de forma de satisfacer cierta demanda, $d_t \geq 0$, mientras se minimizan costos fijos de producción, f_t , costos unitarios de producción, p_t , y costos de depósito, h_t , para el almacenamiento de la producción no vendida al finalizar cada uno de los períodos.

Variables de Decisión

- x_t es la cantidad producida en el período t ,
- s_t es el inventario al final del período t ,
- $y_t = 1$ si se produce en el período t , $y_t = 0$ en otro caso.

Restricciones

- equilibrio de inventario según períodos:

$$\begin{aligned} s_{t-1} + x_t &= d_t + s_t, & t = 1, \dots, n, \\ s_0 &= 0 \end{aligned}$$

- activación de producción

Dado que no hay una cota de producción se asume un valor grande M para la activación de los costos fijos

$$x_t \leq My_t, \quad t = 1, \dots, n,$$

Función Objetivo

- **minimizar** $\sum_{t=1}^n (p_t x_t + h_t s_t + f_t y_t)$

Problema de determinación de lotes no capacitada (Formulación)

$$\left\{ \begin{array}{l} \min \quad \sum_{t=1}^n (p_t x_t + h_t s_t + f_t y_t) \\ \text{s.a.} \quad s_{t-1} + x_t = d_t + s_t, \quad t = 1, \dots, n \\ \quad \quad x_t \leq M y_t, \quad t = 1, \dots, n \\ \quad \quad s_0 = 0, \\ \quad \quad x_t \geq 0, \quad t = 1, \dots, n \\ \quad \quad s_t \geq 0, \quad t = 1, \dots, n \\ \quad \quad y_t \in \{0, 1\}, \quad t = 1, \dots, n \end{array} \right.$$

Implementación del Problema de determinación de lotes no capacitada (Conjuntos y Parámetros)

```
1 # PROBLEMA DE DETERMINACION DE LOTES NO CAPACITADA
2 #
3
4 set T;
5 /* Periodos */
6
7 param M;
8
9 param f{t in T};
10 /* Costo fijo de producir en el periodo t */
11
12 param p{t in T};
13 /* Costo unitario de produccion en el periodo t, */
14
15 param h{t in T};
16 /* costo unitario de almacenamiento en el periodo t */
17
18 param d{t in T};
19 /* demanda en el periodo t */
```

Implementación del Problema de determinación de lotes no capacitada (Variables y objetivo)

```
21 var x{t in T} >= 0;
22 /* Cantidades a producirse en el periodo t */
23
24 var s{i in (T union {0})} >= 0;
25 /* inventario al final del periodo t */
26
27 var y{t in T}, binary;
28 /* 1 si se produce en el periodo t, 0 en otro caso */

30 minimize costo: sum{t in T} p[t] * x[t] + h[t] * s[t] + f[t] *
    y[t];
31 /* Costo minimo */
```

Implementación del Problema de determinación de lotes no capacitada (Restricciones)

```
33 s.t. equilibrio{t in T}: s[i-1] + x[t] = d[t] + s[t];  
34 /* Equilibrio de inventario */  
35  
36 s.t. activ{t in T}: x[t] <= M * y[t];  
37 /* Activacion de la produccion */  
38  
39 s.t. inicial: s[0] = 0;  
40 /* Inventario inicial */
```

El proyecto GLPK esta disponible en

<http://www.gnu.org/software/glpk>

GLPK para MS-Windows esta disponible en

<http://winglpk.sourceforge.net/>