***TRABAJO PRÁCTICO – Entrega 2***

**71.14 - MODELOS Y OPTIMIZACIÓN I**

17/10/2013

**UNIVERSIDAD DE BUENOS AIRES**

Grupo Nro. 3

Diego Costa (78189)

Nahuel Persia (90772)

Ariel Liguori (89187)

INDICE

[INDICE 2](#_Toc369797075)

[Enunciado 3](#_Toc369797076)

[Parte I 3](#_Toc369797077)

[Parte II 3](#_Toc369797078)

[Parte III 4](#_Toc369797079)

[PARTE I 5](#_Toc369797080)

[PARTE II 7](#_Toc369797081)

[Set de datos generado 7](#_Toc369797082)

[Primer modelo 8](#_Toc369797083)

[Modelo matemático 8](#_Toc369797084)

[Modelo GLPK 8](#_Toc369797085)

[Set de datos GLPK (modelo I) 9](#_Toc369797086)

[Segundo modelo 10](#_Toc369797087)

[Modelo matemático 10](#_Toc369797088)

[Modelo GLPK 10](#_Toc369797089)

[Set de datos GLPK (modelo II) 11](#_Toc369797090)

[Tercer modelo 12](#_Toc369797091)

[Modelo matemático 12](#_Toc369797092)

[Modelo GLPK 12](#_Toc369797093)

[Set de datos GLPK (modelo III) 13](#_Toc369797094)

[Ejecuciones de GLPK 14](#_Toc369797095)

[Modelo I 14](#_Toc369797096)

[Modelo II 16](#_Toc369797097)

[Modelo III 16](#_Toc369797098)

[Resultados 18](#_Toc369797099)

[PARTE III 19](#_Toc369797100)

# 

# Enunciado

## Parte I

La famosa empresa "Dinero Seguro" se dedica al transporte de caudales y ha decidido contratarte ya que necesita de tu ingenio para resolver su problema.

En esta oportunidad tiene la misión de asegurarse que uno de sus camiones cumpla su recorrido por los bancos de forma exitosa. Dicho camión debe salir de una de las sedes de la empresa, pasar por 10 bancos y retornar a esa misma sede. Por razones de seguridad, el camión no podrá detenerse en ningún otro lugar alternativo.

Se sabe que en algunos bancos recaudará dinero de los cajeros (+$) y en otros dejara (-$). Por cierto, los montos ya está+n preestablecidos y se indican a continuación.

Banco Porteño +$A

Banco Del Plata -$B

Banco De Los Andes +$C

Banco Plural +$D

Banco Del Norte -$E

Banco Pampeano -$F

Banco Cooperativo +$G

Banco Sol -$H

Banco República +$I

Banco Vientos del Sur +$J

Asimismo, el camión tiene espacio para transportar hasta $MAX\_DINERO y debe asegurarse de tener dinero disponible para dejar en el caso en que le toque un banco con esa necesidad.

Las distancias entre dos bancos cualesquiera (medidas en kilómetros) son datos fijos. También lo es la distancia entre la sede de la empresa y cada banco.

Analizar el problema y realizar propuestas para su resolución en clase. Redactar objetivo, hipótesis y explicar alguna propuesta de resolución elegida

## Parte II

Modelización lineal y heurísticas.

A partir del problema presentado en la Parte I, se pide:

1. Modelo lineal matemático y corrida en alguna de las herramientas sugeridas por la cátedra (se recomiendo GLPK o CPLEX) eligiendo un set de datos factible (Entrega 17/10).

2. Corridas de otros modelos vistos en clase que resuelven el problema. Comparación entre cada corrida y resultado (Entrega 17/10).

## Parte III

Análisis de sensibilidad

La cooperativa "La tranquilísima" ha decidido enfrentar la crisis del campo

y salir a competir con los gigantes de la industria láctea. Para esto, ha decidido

lanzar 2 nuevos productos, yogur "ligerísimo" y queso "picadille".

Para producir 1 kg de yogurt, se requieren 0.5 litros de leche, para producir 1

kg. de queso, se requiere 1.5 litros de leche. Se dispone de 5.000 litros de leche

mensuales.

Para que el yogurt sea competitivo, rico en nutrientes y comestible, cada kg

debe tener 5 gr de endulzante. Se disponen de 50 Kg. de endulzante mensuales.

En cuanto a nutrientes enriquecedores, el yogurt tiene que tener un 25% de

su peso y el queso un 35% de su peso. Se dispone de 1.500 kg. de nutrientes

enriquecedores mensualemente.

Para no perder mercado contra los competidores, debemos producir mas de

500 kg de Yogurt al mes. El precio unitario del yogurt x 330 gr es $2.50 y el

del queso x 500 gr es $ 12.

Realizar el modelo matemático y correrlo en LINDO. (Entrega 12/09).

# PARTE I

**Objetivos**:

Definir el recorrido de un camión de la empresa de transportes de caudales “Dinero Seguro”. Dicho camión deberá salir de una sede de la Empresa y al cabo del día de trabajo haber pasado por 10 bancos y retornar a la empresa. El camión depositará o recaudará montos preestablecidos en cada banco y podrá realizarlo siempre que no sobrepase un monto máximo en su interior, el cual es preestablecido.

**Hipótesis**:

* Los montos preestablecidos de depósito o recaudación en los bancos deberán ser respetados (Lo que se baja del camión entra en el banco y lo que sale del banco entra en el camión, sin perdidas).
* El camión no sufrirá desperfectos, ni hurtos.
* Los montos preestablecidos de carga y descarga no pueden superar el máximo que puede transportar el camión.
* El camión sólo podrá pasar una única vez por cada banco.
* El camión sólo podrá realizar una operación en el banco, o carga o descarga dinero.
* No se consideran problemas de tráfico ni desvíos que demoren el traslado de caudales, ni que afecten la distancia prestablecida entre bancos.
* No se considera el tiempo perdido en la carga o descarga de caudales.
* El camión no puede detenerse en ningún lugar que no sea un banco.
* De modificarse los montos preestablecidos de los bancos, el modelo se deberá volver a resolver con los nuevos datos.
* El camión podrá salir con caudales o regresar con caudales para poder satisfacer las necesidades de los bancos.
* El camión como mínimo podrá quedar sin dinero.

**Una posible solución**:

Si se analiza en forma de grafo, siendo los bancos y la sede desde la cual se parte (y se termina) los nodos, y en el cual, cada camino este ponderado por la distancia entre destinos y las formas de conectar los distintos bancos y la empresa, se podrá realizar un análisis partiendo desde la sede, buscando los bancos más cercanos, y terminando en la misma sede.

Al buscar el banco más cercano, decidir si se puede depositar o extraer, el que corresponda (validando que la operación a realizar no me haga superar el máximo permitido por el camión en caso de extracción o que tenga dinero suficiente en el camión para realizarla, en caso de depósito), o seguir analizando el siguiente más cercano.

Otra posible solución sería realizar una selección al azar del siguiente banco a visitar

También se podría realizar un análisis por fuerza bruta, y luego seleccionar la mejor solución respecto a un criterio que seleccionemos (km. Recorridos, por ejemplo).

# PARTE II

Desarrollaremos tres modelos matemáticos para la resolución del problema del viajante planteado en el enunciado. Para ello hemos generado un set de datos común a todos ellos con el fin de poder evaluar las corridas vía simulación en GLPK.

## Set de datos generado

Set de datos de distancias (en km.):

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | O | A | B | C | D | E | F | G | H | I | J |
| O | - | 5 | 2 | 4 | 6 | 3 | 5 | 4 | 5 | 7 | 9 |
| A | 5 | - | 4 | 2 | 4 | 3 | 3 | 4 | 5 | 4 | 4 |
| B | 2 | 4 | - | 3 | 5 | 4 | 3 | 1 | 4 | 2 | 3 |
| C | 4 | 2 | 3 | - | 3 | 4 | 4 | 5 | 5 | 4 | 4 |
| D | 6 | 4 | 5 | 3 | - | 4 | 5 | 5 | 3 | 2 | 4 |
| E | 3 | 3 | 4 | 4 | 4 | - | 2 | 6 | 5 | 6 | 7 |
| F | 5 | 3 | 3 | 4 | 5 | 2 | - | 3 | 3 | 5 | 5 |
| G | 4 | 4 | 1 | 5 | 5 | 6 | 3 | - | 4 | 4 | 5 |
| H | 5 | 5 | 4 | 5 | 3 | 5 | 3 | 4 | - | 3 | 3 |
| I | 7 | 4 | 2 | 4 | 2 | 6 | 5 | 4 | 3 | - | 4 |
| J | 9 | 4 | 3 | 4 | 4 | 7 | 5 | 5 | 3 | 4 | - |

Referencias y Montos:

0: Casa Central

A: Banco Porteño (+ $20.000)

B: Banco Del Plata (- $5.000)

C: Banco de Los Andes (+ $20.000)

D: Banco Plural (+ $5.000)

E: Banco Del Norte (- $15.000)

F: Banco Pampeano (- $10.000)

G: Banco Cooperativo (- $5.000)

H: Banco Sol (- $5.000)

I: Banco República (+ $5.000)

J: Banco Vientos del Sur (+ $5.000)

MAX\_DINERO = $200.000

## Primer modelo

Para esta aproximación se plantea un problema de viajante estándar, se plantean las ecuaciones para eliminación de sub-tours y se agregar la acumulación de dinero que lleva el camión agregando una variable bivalente “visant” que indicará si la ciudad i se visitó antes que la j

### Modelo matemático

*Variable tramo, bivalente que indicará si existe tramo directo entre las ciudades i, j.*

*Variable visant anteriormente descripta.*

*Variable montocam, entera, que indicará la cantidad de dinero que posee el camión al estar en la ciudad i.*

*Variable orden para indicar las U del viajante, es decir el orden de recorrido adoptado.*

Plantearemos:

### Modelo GLPK

# Conjuntos

set BANCOS;

set NODOS;

# Parametros

param DIST{i in NODOS, j in NODOS};

param MONTOS{i in NODOS};

param M;

#Definicion de variables

var tramo{i in NODOS, j in NODOS} binary;

var orden{i in BANCOS} integer >= 0;

var visant{i in BANCOS, j in BANCOS} binary;

var montocam{i in BANCOS} >= 0;

/\* Funcional \*/

minimize z: sum{i in NODOS, j in NODOS} DIST[i,j] \* tramo[i,j];

/\* Restricciones \*/

# Entra y Sale de todos

s.t. dest{j in NODOS}: sum{i in NODOS: i <> j} tramo[i,j] = 1;

s.t. orig{i in NODOS}: sum{j in NODOS: j <> i} tramo[i,j] = 1;

# Sin SubTour

s.t. orde{i in BANCOS, j in BANCOS: i <> j}: orden[i]-orden[j]+(10\*tramo[i,j])<=9;

s.t. estaantes{i in BANCOS, j in BANCOS: i <> j}: orden[i] - (100 \* (1 - visant[i, j])) <= orden[j];

s.t. estadespues{i in BANCOS, j in BANCOS: i <> j}: orden[j] <= orden[i] + (100 \* visant[i, j]);

s.t. carga{j in BANCOS}: sum{i in BANCOS} MONTOS[i] \* visant[i, j] = montocam[j];

s.t. tope{i in BANCOS}: montocam[i] <= 200000;

s.t. piso{i in BANCOS}: montocam[i] >=0;

end;

### Set de datos GLPK (modelo I)

set NODOS := O A B C D E F G H I J ;

set BANCOS := A B C D E F G H I J ;

param MONTOS :=

O 0

A 20000

B -5000

C 20000

D 5000

E -15000

F -10000

G -5000

H -5000

I 5000

J 5000;

param DIST: O A B C D E F G H I J:=

O 0 5 2 4 6 3 5 4 5 7 9

A 5 0 4 2 4 3 3 4 5 4 4

B 2 4 0 3 5 4 3 1 4 2 3

C 4 2 3 0 3 4 4 5 5 4 4

D 6 4 5 3 0 4 5 5 3 2 4

E 3 3 4 4 4 0 2 6 5 6 7

F 5 3 3 4 5 2 0 3 3 5 5

G 4 4 1 5 5 6 3 0 4 4 5

H 5 5 4 5 3 5 3 4 0 3 3

I 7 4 2 4 2 6 5 4 3 0 4

J 9 4 3 4 4 7 5 5 3 4 0;

end;

## Segundo modelo

En este caso se definirá una variable bivalente “visord i,j” que indicará si se ha visitado la región i en el orden j.

### Modelo matemático

Plantearemos:

### Modelo GLPK

# Conjuntos

set BANCOS;

set NODOS;

set ORDENES;

# Parametros

param DIST{i in NODOS, j in NODOS};

param MONTOS{i in NODOS};

param M;

#Definicion de variables

var tramo{i in NODOS, j in NODOS} binary;

var orden{i in BANCOS} integer >= 0;

var visord{i in BANCOS, j in ORDENES} binary;

var montocam{i in ORDENES} >= 0;

/\* Funcional \*/

minimize z: sum{i in NODOS, j in NODOS} DIST[i,j] \* tramo[i,j];

/\* Restricciones \*/

# Entra y Sale de todos

s.t. dest{j in NODOS}: sum{i in NODOS: i <> j} tramo[i,j] = 1;

s.t. orig{i in NODOS}: sum{j in NODOS: j <> i} tramo[i,j] = 1;

# Sin SubTour

s.t. orde{i in BANCOS, j in BANCOS: i <> j}: orden[i]-orden[j]+(10\*tramo[i,j])<=9;

s.t. ord\_unico{i in BANCOS}: sum{j in ORDENES: j<>0 } visord[i,j] = 1;

s.t. region\_ord{i in BANCOS}: sum{j in ORDENES: j<>0} j\*visord[i,j] = orden[i];

s.t. inicial: montocam[0]=0;

s.t. carga{j in ORDENES: j<>0}: montocam[j-1] + sum{i in BANCOS} MONTOS[i] \* visord[i, j] = montocam[j];

s.t. tope{i in ORDENES: i<>0}: montocam[i] <= 200000;

s.t. piso{i in ORDENES: i<>0}: montocam[i] >=0;

end;

### Set de datos GLPK (modelo II)

Idem modelo I, solo se agrega:

set ORDENES := 0 1 2 3 4 5 6 7 8 9 10 ;

## 

## Tercer modelo

Esta última variación propuesta elimina las necesidades de modelizar vía bivalentes quién ha de estar antes inmediatamente y para ello sobrecarga en la resolución del modelo la elección agregando una restricción ligada a una nueva y única variable bivalente “inidic” la cual indicará si el valor del monto del camión en la ciudad j se corresponde al valor de la ciudad i mas el delta de i a j.

### Modelo matemático

Plantearemos:

Esto se reflejará en el modelo como:

### Modelo GLPK

# Conjuntos

set BANCOS;

set NODOS;

# Parametros

param DIST{i in NODOS, j in NODOS};

param MONTOS{i in NODOS};

param M;

#Definicion de variables

var tramo{i in NODOS, j in NODOS} binary;

var orden{i in BANCOS} integer >= 0;

var indic{i in BANCOS, j in BANCOS} binary;

var montocam{i in BANCOS} >= 0;

/\* Funcional \*/

minimize z: sum{i in NODOS, j in NODOS} DIST[i,j] \* tramo[i,j];

/\* Restricciones \*/

# Entra y Sale de todos

s.t. dest{j in NODOS}: sum{i in NODOS: i <> j} tramo[i,j] = 1;

s.t. orig{i in NODOS}: sum{j in NODOS: j <> i} tramo[i,j] = 1;

# Sin SubTour

s.t. orde{i in BANCOS, j in BANCOS: i <> j}: orden[i]-orden[j]+(10\*tramo[i,j])<=9;

s.t. ord\_unico{i in BANCOS}: sum{j in BANCOS: i<>j } indic[i,j] = 1;

#s.t. carga{j in BANCOS}: sum{i in BANCOS} MONTOS[i] \* visant[i, j] = montocam[j];

s.t. sol{i in BANCOS, j in BANCOS: i <> j}: -1000000000\*(1-indic[j,i]) + (montocam[i] + MONTOS[i]) <= montocam[j];

s.t. soldos{i in BANCOS, j in BANCOS: i <> j}: montocam[j]<= (1-indic[j,i])\*10000000 + (montocam[i] + MONTOS[i]);

#s.t. carga{j in BANCOS}: sum{i in BANCOS} MONTOS[i] \* visant[i, j] = montocam[j];

s.t. tope{i in BANCOS}: montocam[i] <= 200000;

s.t. piso{i in BANCOS}: montocam[i] >=0;

end;

### Set de datos GLPK (modelo III)

Idem modelo I.

## Ejecuciones de GLPK

### Modelo I

Problem: forma1

Rows: 323

Columns: 230 (220 integer, 210 binary)

Non-zeros: 1270

Status: INTEGER OPTIMAL

Objective: z = 29 (MINimum)

No. Row name Activity Lower bound Upper bound

------ ------------ ------------- ------------- -------------

1 z 29

…

1 tramo[O,A] \* 0 0 1

2 tramo[O,B] \* 0 0 1

3 tramo[O,C] \* 0 0 1

4 tramo[O,D] \* 0 0 1

5 tramo[O,E] \* 1 0 1

6 tramo[O,F] \* 0 0 1

7 tramo[O,G] \* 0 0 1

8 tramo[O,H] \* 0 0 1

9 tramo[O,I] \* 0 0 1

10 tramo[O,J] \* 0 0 1

11 tramo[A,O] \* 0 0 1

12 tramo[A,B] \* 0 0 1

13 tramo[A,C] \* 1 0 1

14 tramo[A,D] \* 0 0 1

15 tramo[A,E] \* 0 0 1

16 tramo[A,F] \* 0 0 1

17 tramo[A,G] \* 0 0 1

18 tramo[A,H] \* 0 0 1

19 tramo[A,I] \* 0 0 1

20 tramo[A,J] \* 0 0 1

21 tramo[B,O] \* 1 0 1

22 tramo[B,A] \* 0 0 1

23 tramo[B,C] \* 0 0 1

24 tramo[B,D] \* 0 0 1

25 tramo[B,E] \* 0 0 1

26 tramo[B,F] \* 0 0 1

27 tramo[B,G] \* 0 0 1

28 tramo[B,H] \* 0 0 1

29 tramo[B,I] \* 0 0 1

30 tramo[B,J] \* 0 0 1

31 tramo[C,O] \* 0 0 1

32 tramo[C,A] \* 0 0 1

33 tramo[C,B] \* 0 0 1

34 tramo[C,D] \* 1 0 1

35 tramo[C,E] \* 0 0 1

36 tramo[C,F] \* 0 0 1

37 tramo[C,G] \* 0 0 1

38 tramo[C,H] \* 0 0 1

39 tramo[C,I] \* 0 0 1

40 tramo[C,J] \* 0 0 1

41 tramo[D,O] \* 0 0 1

42 tramo[D,A] \* 0 0 1

43 tramo[D,B] \* 0 0 1

44 tramo[D,C] \* 0 0 1

45 tramo[D,E] \* 0 0 1

46 tramo[D,F] \* 0 0 1

47 tramo[D,G] \* 0 0 1

48 tramo[D,H] \* 0 0 1

49 tramo[D,I] \* 1 0 1

50 tramo[D,J] \* 0 0 1

51 tramo[E,O] \* 0 0 1

52 tramo[E,A] \* 1 0 1

53 tramo[E,B] \* 0 0 1

54 tramo[E,C] \* 0 0 1

55 tramo[E,D] \* 0 0 1

56 tramo[E,F] \* 0 0 1

57 tramo[E,G] \* 0 0 1

58 tramo[E,H] \* 0 0 1

59 tramo[E,I] \* 0 0 1

60 tramo[E,J] \* 0 0 1

61 tramo[F,O] \* 0 0 1

62 tramo[F,A] \* 0 0 1

63 tramo[F,B] \* 0 0 1

64 tramo[F,C] \* 0 0 1

65 tramo[F,D] \* 0 0 1

66 tramo[F,E] \* 0 0 1

67 tramo[F,G] \* 1 0 1

68 tramo[F,H] \* 0 0 1

69 tramo[F,I] \* 0 0 1

70 tramo[F,J] \* 0 0 1

71 tramo[G,O] \* 0 0 1

72 tramo[G,A] \* 0 0 1

73 tramo[G,B] \* 1 0 1

74 tramo[G,C] \* 0 0 1

75 tramo[G,D] \* 0 0 1

76 tramo[G,E] \* 0 0 1

77 tramo[G,F] \* 0 0 1

78 tramo[G,H] \* 0 0 1

79 tramo[G,I] \* 0 0 1

80 tramo[G,J] \* 0 0 1

81 tramo[H,O] \* 0 0 1

82 tramo[H,A] \* 0 0 1

83 tramo[H,B] \* 0 0 1

84 tramo[H,C] \* 0 0 1

85 tramo[H,D] \* 0 0 1

86 tramo[H,E] \* 0 0 1

87 tramo[H,F] \* 1 0 1

88 tramo[H,G] \* 0 0 1

89 tramo[H,I] \* 0 0 1

90 tramo[H,J] \* 0 0 1

91 tramo[I,O] \* 0 0 1

92 tramo[I,A] \* 0 0 1

93 tramo[I,B] \* 0 0 1

94 tramo[I,C] \* 0 0 1

95 tramo[I,D] \* 0 0 1

96 tramo[I,E] \* 0 0 1

97 tramo[I,F] \* 0 0 1

98 tramo[I,G] \* 0 0 1

99 tramo[I,H] \* 0 0 1

100 tramo[I,J] \* 1 0 1

101 tramo[J,O] \* 0 0 1

102 tramo[J,A] \* 0 0 1

103 tramo[J,B] \* 0 0 1

104 tramo[J,C] \* 0 0 1

105 tramo[J,D] \* 0 0 1

106 tramo[J,E] \* 0 0 1

107 tramo[J,F] \* 0 0 1

108 tramo[J,G] \* 0 0 1

109 tramo[J,H] \* 1 0 1

110 tramo[J,I] \* 0 0 1

221 montocam[A] 5000 0

222 montocam[B] 20000 0

223 montocam[C] 5000 0

224 montocam[D] 25000 0

225 montocam[E] 0 0

226 montocam[F] 35000 0

227 montocam[G] 25000 0

228 montocam[H] 40000 0

229 montocam[I] 30000 0

230 montocam[J] 35000 0

Con esta información es posible determinar el orden de recorrido (variable tramo i,j) y los montos del camión en cada ciudad (variable montocam i)

Se detalla a continuación la salida de GLPK

Integer optimization begins...

Gomory's cuts enabled

MIR cuts enabled

Cover cuts enabled

Clique cuts enabled

Constructing conflict graph...

**Conflict graph has 110 + 0 = 110 vertices**

+ 114: mip = not found yet >= -inf (1; 0)

Cuts on level 0: gmi = 7;

Cuts on level 50: gmi = 7;

+ 2479: >>>>> 3.100000000e+001 >= 2.800000000e+001 9.7% (97; 2)

Cuts on level 41: gmi = 15; mir = 1; cov = 2;

+ 23658: >>>>> 2.900000000e+001 >= 2.900000000e+001 < 0.1% (410; 207)

+ 23658: mip = 2.900000000e+001 >= tree is empty 0.0% (0; 1127)

INTEGER OPTIMAL SOLUTION FOUND

**Time used: 5.7 secs**

Memory used: 1.8 Mb (1919599 bytes)

Writing MIP solution to `forma1.sol'...

>Exit code: 0 Time: 6.153

### Modelo II

A continuación se detalla la salida de GLPK:

Integer optimization begins...

Gomory's cuts enabled

MIR cuts enabled

Cover cuts enabled

Clique cuts enabled

Constructing conflict graph...

**Conflict graph has 210 + 0 = 210 vertices**

+ 97: mip = not found yet >= -inf (1; 0)

Cuts on level 0: gmi = 4;

+ 35237: mip = not found yet >= 2.900000000e+001 (678; 63)

+ 67188: mip = not found yet >= 2.900000000e+001 (1336; 154)

Cuts on level 18: gmi = 12; cov = 1;

+ 82889: >>>>> 2.900000000e+001 >= 2.900000000e+001 < 0.1% (1677; 207)

+ 82889: mip = 2.900000000e+001 >= tree is empty 0.0% (0; 3767)

INTEGER OPTIMAL SOLUTION FOUND

**Time used: 12.8 secs**

Memory used: 3.0 Mb (3168781 bytes)

Writing MIP solution to `forma2.sol'...

>Exit code: 0 Time: 13.786

### Modelo III

A continuación se detalla la salida de GLPK:

Integer optimization begins...

Gomory's cuts enabled

MIR cuts enabled

Cover cuts enabled

Clique cuts enabled

Constructing conflict graph...

**Conflict graph has 200 + 0 = 200 vertices**

+ 95: mip = not found yet >= -inf (1; 0)

Cuts on level 0: gmi = 7; mir = 2;

Cuts on level 26: gmi = 7; mir = 2;

+ 448: >>>>> 3.000000000e+001 >= 2.800000000e+001 6.7% (27; 0)

Cuts on level 29: gmi = 11; mir = 2;

+ 984: >>>>> 2.900000000e+001 >= 2.900000000e+001 0.0% (41; 46)

+ 984: mip = 2.900000000e+001 >= tree is empty 0.0% (0; 129)

INTEGER OPTIMAL SOLUTION FOUND

**Time used: 0.5 secs**

Memory used: 1.0 Mb (998716 bytes)

Writing MIP solution to `forma3.sol'...

## Resultados

Como se puede observar a lo largo del desarrollo del trabajo, los tres modelos han resuelto el problema planteado, sin embargo debemos destacar que el primer modelo y el tercero lo han realizado en menor tiempo que el segundo. También hay que mencionar que los grafos de caminos generados no siempre han sido los mismos (había múltiples caminos con mismo peso –menor distancia- y el modelo ha encontrado diversos caminos que satisfacían esta condición) pero siempre las soluciones cumplían con todas las restricciones planteadas.

# PARTE III

**Objetivos**:

Determinar la producción queso y de yogurt (en kilogramos) considerando las restricciones de litros de leche disponibles, la cantidad de nutrientes y la producción mínima para maximizar las ganancias durante un período de tiempo t.

**Hipótesis**:

* Un litro de leche equivale a un kilogramo en el producto final (queso o yogurt).
* Se puede vender el queso o el yogurt en cualquier cantidad y el precio es proporcional al peso.
* En el proceso de producción no hay pérdidas de ingredientes.
* En caso de estar involucrada alguna máquina para el proceso de producción, la misma no falla y hay disponibilidad para hacer uso de ella.
* Hay disponibilidad de leche y nutrientes en todo el proceso productivo.
* El valor indicado de venta de yogurt y de queso, es la ganancia neta ante el mismo y no deben considerarse otros costos.
* Se cuenta con mano de obra para realizar la producción

**Entrada:**

MAX 7.575 XY + 240 XQ

SUBJECT TO

LECHE)0.5 XY + 1.5 XQ <= 5000

0.005 XY <= 50

NUTRIENTESY)NY - 0.000625 XY >= 0

NUTRIENTESQ)NQ - 0.525 XQ >= 0

NY + NQ <= 1500

! RESTRICCIONES

XY >= 500

**Salida:**

Global optimal solution found.

Objective value:                              696164.9

Infeasibilities:                               0.000000

Total solver iterations:                     2

Elapsed runtime seconds:               0.02

Model Class:                                    LP

Total variables:                       4

Nonlinear variables:                 0

Integer variables:                     0

Total constraints:                    7

Nonlinear constraints:             0

Total nonzeros:                      12

  Nonlinear nonzeros:                   0

                                Variable           Value        Reduced Cost

                                      XY        1433.692            0.000000

                                      XQ        2855.436            0.000000

                                      NY       0.8960573            0.000000

                                      NQ        1499.104            0.000000

Row    Slack or Surplus      Dual Price

1        696164.9            1.000000

LECHE        0.000000            14.63082

3        42.83154            0.000000

NUTRIENTESY        0.000000           -415.3405

NUTRIENTESQ        0.000000           -415.3405

6        0.000000            415.3405

7        933.6918            0.000000