

Ejercicio 2.22

[71.14] Modelos y Optimización I
Curso 4
2C 2021

Alumno:	Grassano, Bruno
Número de padrón:	103855
Email:	bgrassano@fi.uba.ar

Índice

1. Enunciado	2
2. Análisis de la situación problemática	3
3. Objetivo	4
4. Hipótesis y supuestos	4
5. Definición de variables	4
6. Modelo de programación lineal	5
6.1. Funcional	5
6.2. Restricciones	6
7. Resolución por software	7
8. Informe de la solución óptima	10

1. Enunciado

Un afamado restorán de la zona de Palermo, tiene que entregar por lo menos 48 Kg de comida el sábado y 40 Kg el domingo, para sendos banquetes tiene 15 empleados que trabajan hasta 8 hs cada uno, a los que les paga \$0,2 la hora (si no trabajan no les paga) y que hacen alguna de las siguiente tareas: comprar alimentos, preparar alimentos y cocinar. Un mismo empleado no puede cocinar dos días seguidos.

Un empleado puede comprar 10 Kg de alimentos por día. A cada empleado le lleva media hora cocinar o preparar 1 Kg de alimentos. Por un convenio laboral no puede haber menos de cuatro empleados que cocinen cada día.

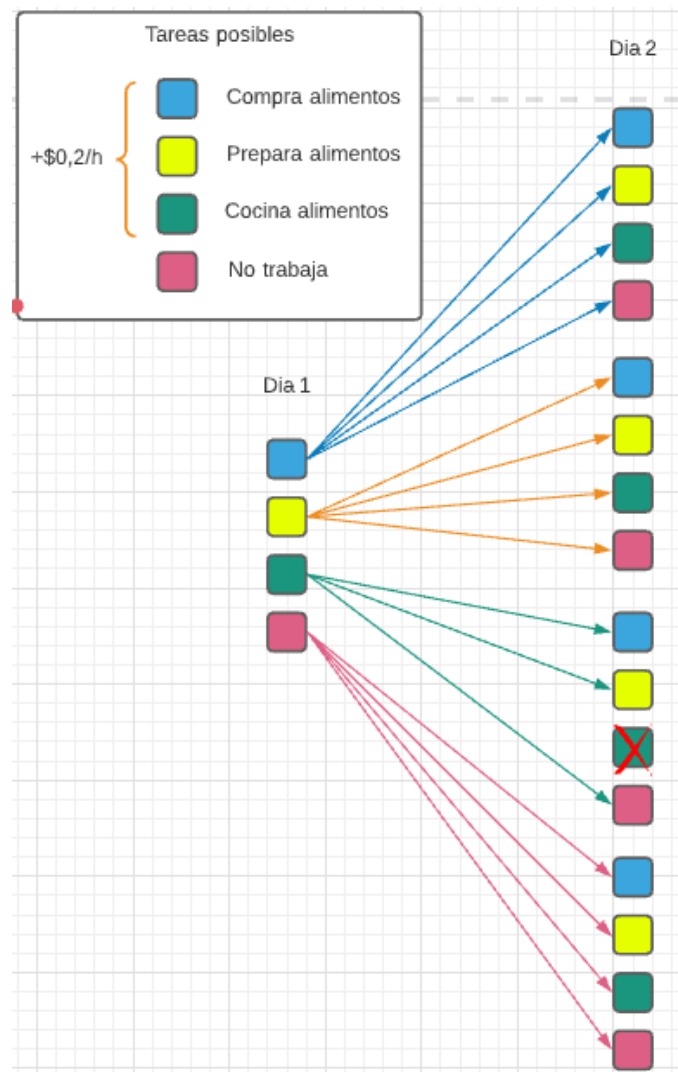
Si el domingo hay más empleados comprando alimentos que el sábado, el restorán pagará un plus de \$1 por empleado de más. Sin embargo, esa cantidad de empleados en exceso no puede ser superior a cuatro. Si el domingo hay menos empleados comprando alimentos que el sábado, el restorán deberá pagar \$2 por empleado de menos, salvo que el domingo haya más de cuatro empleados cocinando.

Análisis previo: comenzar la resolución del ejercicio, realizando un esquema que describa la situación

Opcional: Obtener una solución óptima para este problema utilizando un software de resolución de problemas lineales (LINDO).

2. Análisis de la situación problemática

- Se nota que es un problema con multiperiodo, entregas para el sábado, y otras para el domingo.
- Destacar también que están relacionados ambos días, ya que un empleado no puede cocinar dos días seguidos.
- Se notan condiciones débiles para el domingo respecto de la cantidad de empleados.
- El siguiente es un esquema del problema:



3. Objetivo

Determinar la mejor forma de asignar las tareas a los empleados durante cada día para minimizar los costos durante los dos días.

4. Hipótesis y supuestos

1. Todos los empleados están disponibles si son requeridos.
2. Los empleados no pueden negarse a realizar la tarea que les corresponda.
3. Los costos que conlleva emplearlos son precisos.
4. La comida no se estropea.
5. No se tiene un stock inicial de alimentos.
6. La compra de alimentos por cada empleado es exacta.
7. El transporte de los alimentos al restaurante es despreciable.
8. Los empleados que cocinen o preparen alimentos siempre realizan la cantidad indicada en una media hora.
9. No hay más costos aparte de los indicados en el enunciado.
10. No hay más convenios laborales aparte del indicado.
11. Se puede guardar alimentos y comida de un día para el otro.
12. Los alimentos y comida no se estropean.
13. Se tiene el espacio para almacenar los alimentos y comida, además de prepararlos y cocinarlos.
14. La comida que se puede entregar tiene que ser preparada o cocinada. No se prepara primero y después se cocina.
15. Los empleados que compren alimentos lo hacen en todo un día laboral (8 horas).

5. Definición de variables

**Con tipos y unidades*

- COMCOM: Cantidad de empleados que compran el primer día y compran el segundo. (unidad/días) (entera)
- COMPRE: Cantidad de empleados que compran el primer día y preparan el segundo. (unidad/días) (entera)
- COMCOC: Cantidad de empleados que compran el primer día y cocinan el segundo. (unidad/días) (entera)
- COMNOT: Cantidad de empleados que compran el primer día y no trabajan el segundo. (unidad/días) (entera)
- PRECOM: Cantidad de empleados que preparan el primer día y compran el segundo. (unidad/días) (entera)
- PREPRE: Cantidad de empleados que preparan el primer día y preparan el segundo. (unidad/días) (entera)

- PRECOC: Cantidad de empleados que preparan el primer día y cocinan el segundo. (unidad/días) (entera)
- PRENOT: Cantidad de empleados que preparan el primer día y no trabajan el segundo. (unidad/días) (entera)
- COCCOM: Cantidad de empleados que cocinan el primer día y compran el segundo. (unidad/días) (entera)
- COCPRE: Cantidad de empleados que cocinan el primer día y preparan el segundo. (unidad/días) (entera)
- COCNOT: Cantidad de empleados que cocinan el primer día y no trabajan el segundo. (unidad/días) (entera)
- NOTCOM: Cantidad de empleados que no trabajan el primer día y compran el segundo. (unidad/días) (entera)
- NOTPRE: Cantidad de empleados que no trabajan el primer día y preparan el segundo. (unidad/días) (entera)
- NOTCOC: Cantidad de empleados que no trabajan el primer día y cocinan el segundo. (unidad/días) (entera)
- NOTNOT: Cantidad de empleados que no trabajan el primer día y no trabajan el segundo. (unidad/días) (entera)
- EXCESO: Cantidad de empleados comprando alimento de mas el domingo. (unidad/días) (entera)
- DEFECTO: Cantidad de empleados comprando alimento de menos el domingo. (unidad/días) (entera)
- ALIMSTOCK: Cantidad de alimento en stock para el día siguiente. (kg/días) (continua)
- COMISTOCK: Cantidad de comida en stock para el día siguiente. (kg/días) (continua)

6. Modelo de programación lineal

**Indicando en cada restricción o grupo de restricciones la función que cumplen.*

6.1. Funcional

Buscamos minimizar los costos:

$$3,2 \cdot (COMCOM + COMPRE + COMCOC + PRECOM + PREPRE + PRECOC + COCCOM + COCPRE) + 1,6 \cdot (COMNOT + PRENOT + COCNOT + NOTCOM + NOTPRE + NOTCOC) + 1 \cdot EXCESO + 2 \cdot DEFECTO$$

Donde:

- Por trabajar dos días: $3,2 = 0,2 \frac{\$}{hs} \cdot 2días \cdot 8hs$
- Por trabajar un día: $1,6 = 0,2 \frac{\$}{hs} \cdot 8hs$

6.2. Restricciones

Comenzamos planteando el limite de empleados que tenemos:

- $COMCOM + COMPRE + COMCOC + PRECOM + PREPRE + PRECOC + COCCOM + COCPRE + COMNOT + PRENOT + COCNOT + NOTCOM + NOTPRE + NOTCOC + NOTNOT = 15$

Planteamos la cantidad de comida que requerimos entregar:

- Para el primer día: $16 \cdot (PRECOM + PREPRE + PRECOC + PRENOT + COCCOM + COCPRE + COCNOT) \geq 48 + COMISTOCK^1$
- Para el segundo día: $16 \cdot (COMPRE + COMCOC + PREPRE + PRECOC + COCPRE + NOTPRE + NOTCOC) + COMISTOCK \geq 40$

Estamos limitados también por la cantidad de alimento que se compra para preparar o cocinar.

- Para el primer día: $10 \cdot (COMCOM + COMPRE + COMCOC + COMNOT) \geq 16 \cdot (PRECOM + PREPRE + PRECOC + PRENOT + COCCOM + COCPRE + COCNOT) + ALIMSTOCK$
- Para el segundo día: $10 \cdot (COMCOM + PRECOM + COCCOM + NOTCOM) + ALIMSTOCK \geq 16 \cdot (COMPRE + COMCOC + PREPRE + PRECOC + COCPRE + NOTPRE + NOTCOC) + COMISTOCK$

Nos quedan las restricciones de cuantos empleados puede haber cocinando, y de los limites del domingo:

- Para los cocineros el primer día: $COCCOM + COCPRE + COCNOT \geq 4$
- Para los cocineros el segundo día: $COMCOC + PRECOC + NOTCOC \geq 4$
- Restricción de las compras: $(COMCOM + PRECOM + COCCOM + NOTCOM) - (COMCOM + COMPRE + COMCOC + COMNOT) = EXCESO - DEFECTO$
- No puedo tener mas de 4 en exceso: $EXCESO \leq 4$
- Salvo que el domingo haya mas de cuatro empleados cocinando: $COMCOC + PRECOC + NOTCOC - 5 \geq DEFECTO^2$

¹Aclaración del 16: 1kg cada media hora, trabajan 8 horas, en total por día producen 16kg de comida

²No estoy muy seguro de como plantear esta restricción.

7. Resolución por software

El modelo:

```

var COMCOM >=0 integer;
var COMPRE >=0 integer;
var COMCOC >=0 integer;
var COMNOT >=0 integer;

var PRECOM >=0 integer;
var PREPRE >=0 integer;
var PRECOC >=0 integer;
var PRENOT >=0 integer;

var COCCOM >=0 integer;
var COCPRE >=0 integer;
var COCNOT >=0 integer;

var NOTCOM >=0 integer;
var NOTPRE >=0 integer;
var NOTCOC >=0 integer;
var NOTNOT >=0 integer;

var EXCESO >=0 integer;
var DEFECTO >=0 integer;

var ALIMSTOCK >=0;
var COMISTOCK >=0;

minimize z: 3.2 * (COMCOM+COMPRE+COMCOC+PRECOM+PREPRE+PRECOC+COCCOM+COCPRE)
+ 1.6 * (COMNOT+PRENOT+COCNOT+NOTCOM+NOTPRE+NOTCOC) + EXCESO+ 2 * DEFECTO;

s.t. limEmpleados: COMCOM+COMPRE+COMCOC+PRECOM+PREPRE+PRECOC+COCCOM+COCPRE
+COMNOT+PRENOT+COCNOT+NOTCOM+NOTPRE+NOTCOC+NOTNOT = 15;
s.t. comidaSabado: 16 * (PRECOM+PREPRE+PRECOC+PRENOT+COCCOM+COCPRE+COCNOT) >= 48 + COMISTOCK;
s.t. comidaDomingo: 16 * (COMPRE+COMCOC+PREPRE+PRECOC+COCPRE+NOTPRE+NOTCOC) +COMISTOCK >= 40;

s.t. alimentoSabado: 10 * (COMCOM+COMPRE+COMCOC+COMNOT)
>= 16 * (PRECOM+PREPRE+PRECOC+PRENOT+COCCOM+COCPRE+COCNOT) +ALIMSTOCK;
s.t. alimentoDomingo: 10 * (COMCOM+PRECOM+COCCOM+NOTCOM)+ALIMSTOCK
>= 16 * (COMPRE+COMCOC+PREPRE+PRECOC+COCPRE+NOTPRE+NOTCOC) +COMISTOCK;

s.t. cocinaSabado: COCCOM+COCPRE+COCNOT>=4;
s.t. cocinaDomingo: COMCOC+PRECOC+NOTCOC>=4;

s.t. compras: (COMCOM+PRECOM+COCCOM+NOTCOM)-(COMCOM+COMPRE+COMCOC+COMNOT) =EXCESO-DEFECTO;
s.t. limExceso: EXCESO<=4;
s.t. salvoCocinaDomingo: COMCOC+PRECOC+NOTCOC-5>=DEFECTO;
end;

```


Los resultados:

Problem: 2
 Rows: 11
 Columns: 19 (17 integer, 0 binary)
 Non-zeros: 91
 Status: INTEGER OPTIMAL
 Objective: z = 39.4 (MINimum)

No.	Row name	Activity	Lower bound	Upper bound
1	z	39.4		
2	limEmpleados	15	15	=
3	comidaSabado	58	48	
4	comidaDomingo	86	40	
5	alimentoSabado	0	-0	
6	alimentoDomingo	0	-0	
7	cocinaSabado	4	4	
8	cocinaDomingo	5	4	
9	compras	0	-0	=
10	limExceso	1		4
11	salvoCocinaDomingo	5	5	

No.	Column name	Activity	Lower bound	Upper bound
1	COMCOM *	2	0	
2	COMPRE *	0	0	
3	COMCOC *	5	0	
4	COMNOT *	0	0	
5	PRECOM *	0	0	
6	PREPRE *	0	0	
7	PRECOC *	0	0	
8	PRENOT *	0	0	
9	COCCOM *	4	0	
10	COCPRE *	0	0	
11	COCNOT *	0	0	
12	NOTCOM *	2	0	
13	NOTPRE *	0	0	
14	NOTCOC *	0	0	
15	NOTNOT *	2	0	
16	EXCESO *	1	0	
17	DEFECTO *	0	0	
18	ALIMSTOCK	6	0	
19	COMISTOCK	6	0	

Integer feasibility conditions:

KKT.PE: max.abs.err = 1.42e-14 on row 6
 max.rel.err = 8.21e-17 on row 6

High quality

KKT.PB: max.abs.err = 3.20e-14 on row 6

max.rel.err = 3.20e-14 on row 6

High quality

End of output

8. Informe de la solución óptima

La mejor forma de asignar las tareas para minimizar los costos es asignando 2 empleados a que compren ambos días, 5 a que compren el primero y cocinen el segundo, 4 a que cocinen el primero y compren el segundo, dos que solo compren en el segundo, y los otros dos que no trabajen. De esta forma solamente se estaría pagando \$39,4 y entregando 58kg de comida para el primer día, y 86 para el segundo.