Trabajo Practico 2

Zoe Borrone Luca Mazzarello Ignacio Pardo Olivier Saint-Nom

2023-11-03

Introducción

El objetivo del trabajo práctico es plantear un modelo de *lot sizing* para planificar la producción en una fábrica. La planificación es mensual, y consiste en determinar qué cantidad fabricar de cada producto en cada mes, durante los próximos 24 meses. Se tienen cinco productos y se tiene una estimación de la demanda por producto y mes. Suponemos que las cantidades demandadas de todos los productos están en las mismas unidades. La planificación está sujeta a las siguientes restricciones:

1. Cada producto tiene una capacidad de producción máxima de 120 unidades por mes.
2. Los productos se deben fabricar en lotes de 10 unidades cada uno. Es decir, la producción de cada producto en cada mes debe ser un número perteneciente al conjunto {0, 10, 20, 30, . . . , 120}.
3. El depósito tiene una capacidad total de 900 unidades, entre todos los productos. Se puede guardar stock fabricado en un mes para cubrir la demanda de los meses siguientes.
4. Al inicio de la planificación no se tiene stock inicial de ningún producto.
5. La producción de una unidad de cada producto tiene un costo de $370 (el mismo costo por unidad para todos los productos).

El objetivo de la planificación es determinar qué cantidad fabricar de cada producto en cada mes, de modo tal de cumplir la demanda minimizando los costos de fabricación.

Definiciones de la planificación

- Producto : $P = A, B, C, D, E, F$
- Tiempo : $T = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12$
- Demanda del Producto P en el Tiempo T : D_{pt}
- Lotes fabricados del Producto P en el Tiempo T : L_pt

$$
\begin{aligned}
&L_{pt} \in \mathbb{Z}, \forall p \in P, \forall t \in T \\
&L_{pt} >= 0 \\
&L_{pt} <= 12
\end{aligned}
$$

Cantidad de unidades del producto P en el Tiempo T : U_{pt}

$$U_{pt} = L_{pt} * 10$$

Sumatoria de U_{pt} para cada P en el Tiempo $T <= 120$: U_t

$$U_t = \sum_{p \in P}$$
$$U_{pt} \leq 120$$
$$min \sum_{p \in P} \sum_{t \in T} U_{pt} * 370$$

Stock del Producto P en el Tiempo T : S_{pt}

$$S_{pt} = S_{p(t-1)} + U_{pt} - D_{pt}$$
$$S_{pt} >= 0$$
$$\sum_{t \in T} \sum_{p \in P} S_{pt} <= 900$$

Ejercicio 1

El ejercicio 1 consiste en plantear un modelo de programación lineal entera mixta para este problema, con el objetivo de minimizar los costos totales de producción. Para ello, definimos las variables de decisión, la función objetivo y las restricciones del modelo:

Variables de decisión

```
set P := {"A","B","C","D","E"};

set T := {1 .. 24};
set T_ := {0 .. 24};

param D[T*P] := read "tp2.2023_fresco.dat" as "n+";
param C := 370;

var L[P*T] integer >= 0 <= 12;
var S[P*T_] integer >= 0;
```

En el caso de la variable T, esta definida de 1 a 24 porque el transcurso del tiempo son 2 años, es decir, 24 meses. Luego la variable T__ esta definida de 0 a 24 porque se necesita el stock inicial, es decir, el stock del mes anterior al primero.

Función objetivo

```
minimize costo_total: sum <p,t> in P*T: L[p,t] * 10 * C;
```

Buscamos minimizar el costo total de producción, que es la suma de la cantidad de lotes fabricados por el costo de fabricar cada lote (10 unidades por lote, el costo de fabricar una unidad es de $ 370).

Restricciones

Las restricciones que planteamos son las siguientes:

- Stock: el stock del producto en el tiempo t es igual al stock del producto en el tiempo $t-1$ más la cantidad de lotes fabricados en el tiempo t menos la demanda del producto en el tiempo t.
- Stock inicial: el stock inicial de cada producto es 0.
- Stock máximo: la suma del stock de todos los productos en el tiempo t no puede superar las 900 unidades.

```
subto stock:
    forall <p,t> in P*T:
        S[p,t] == S[p,t-1] + L[p,t] * 10 - D[t,p];

subto stock_inicial:
    forall <p> in P:
        S[p,0] == 0;

subto stock_max:
    forall <t> in T:
        sum <p> in P: S[p,t] <= 900;
```

Como solución obtenemos que el costo total es de $ 2904500

Ejercicio 2

En el ejercicio 2, se nos suma una nueva restriccion que es que la produccion total de todos los productos no debe superar las 300 unidades mensuales. Para ello, definimos una nueva restriccion pero mantenemos tanto las función objetivo como las variables de decisión.

Nueva Restricción

La restriccion impuesta es que la suma de las unidades producidas por cada producto no debe superar las 300 unidades por mes.

```
subto limite_unidades:
    forall <t> in T:
        (sum <p> in P: L[p,t] * 10) <= 300;
```

Al correr el código vimos que era infactible cumplir con toda la demanda tras agregar esta restriccion.

Ejercicio 3

Sumandole a la restriccion del ejercicio 2, ahora se nos agrega que podemos tercerizar la fabricacion de hasta 200 unidades por mes, con un costo de $ 540 por unidad. Para ello, cambiamos nuestra función objetivo y agregamos a nuestro modelo, teniendo en cuenta que debemos cumplir toda la demanda de cada mes:

Nuevas variables de decision

Creamos una variable para almacenar las unidades producidas por la fabrica y otra para almacenar las unidades tercerizadas.

```
var U[P*T] integer >= 0;
var F[P*T] integer >= 0; # unidaddes tercerizadas
```

Nueva función objetivo

Agregamos a la funcion objetivo la suma de las unidades tercerizadas por el costo de tercerizarlas.

```
minimize costo_total: sum <p,t> in P*T: L[p,t] * 10 * C + F[p,t] * 540;
```

Nueva restricción

Agregamos la restriccion de que la suma de las unidades tercerizadas no puede superar las 200 unidades por mes.

```
subto limite_uni_tercerizado:
    forall <t> in T:
        (sum <p> in P: F[p,t]) <= 200;
```

El valor objetivo nos queda $3128900

Ejercicio 4

Ahora se nos agrega que si tercerizamos la fabricacion de un producto en un mes, se deben solicitar al menos 20 unidades de ese producto en ese mes. Para ello, agregamos al modelo una nueva variable de decisión y dos nuevas restricciones, manteniendo la función objetivo:

Nuevas variables de decisión

```
var ter[P*T] binary;
```

Esta nueva variable indica si se terceriza (ter[p,t] = 1) o no (ter[p,t] = 0) la fabricacion de un producto en un mes.

Nuevas restricciones

```
subto limite_uni_tercerizado:
    forall <p_,t> in P*T:
        (sum <p> in P: F[p,t]) <= 200 * ter[p_,t];

subto tercerizar_min:
    forall <p,t> in P*T:
        F[p,t] >= 20 * ter[p,t];
```

Para esta nueva retricción, si ter[p,t] = 0, entonces F[p,t] = 0, por lo que no se terceriza la fabricacion de ese producto en ese mes. Si ter[p,t] = 1, entonces F[p,t] >= 20, por lo que se terceriza la fabricacion de ese producto en ese mes y se deben fabricar al menos 20 unidades de ese producto en ese mes. Ademas, la suma de los productos tercerizados no puede superar las 200 unidades por mes.

En este caso el *costo total pasa a ser de $3128900*

Ejercicio 5

El límite de 900 unidades en stock en el depósito es demasiado costoso y necesitamos reducir el espacio disponible en el depósito. Para ello, modificamos el modelo agregando la variable de decision que tambien será la funcion objetivo del ejercicio:

Nueva variable de decisión

```
var stock_m >= 0;
```

Nueva función objetivo

```
minimize stock_maximo_minimo: stock_m;
```

Como resultado obtenemos que la mínima capacidad del depósito que hace que el problema sea factible es 705 (minima capazidad maxima del deposito)

¿Se puede buscar este valor sin hacer búsqueda binaria manualmente?

Es posible y para ello deberiamos hacer un modelo de programacion lineal entera mixta que busque el valor de la capacidad del deposito que hace que el problema sea factible. Para ello, deberiamos agregar una nueva variable de decision que sea la capacidad del deposito y una nueva restriccion que diga que la capacidad del deposito debe ser mayor o igual a la capacidad del deposito minima que hace que el problema sea factible. Luego, la funcion objetivo deberia ser minimizar la capacidad maxima del deposito.