Optimización de rollos de telas en Maine creaciones

Iván Vega Gutiérrez

[1]Centro de Investigación en Matemáticas A.C.
Unidad Aguascalientes
E-mail: ivan.vega@cimat.mx

Índice

I. Introducción

Maine creaciones, es un negocio pequeño de reciente creación que se dedica a la fabricación de vestidos de niño dios en el estado de Oaxaca. Dentro del proceso de la creación de vestidos se requieren distintos tipos de materiales:

- Tela : Se utiliza para la elaboración de las faldas y blusas.
- Galones : Se utiliza para la adornar del vestido
- Elástico: Se utiliza para ajustar los puños de las mangas del vestido.
- Pegamento: Se utiliza pegar los adornos.
- Pedrería: Comúnmente son flores u otras figuras que sirven como adorno de los vestidos.

A pesar de que cada material es indispensable para la elaboración de los vestidos, la materia prima más importante es la tela, ya que la tela es el material que más se ocupa y asimismo el producto que influye más en el costo de la producción. Es por esto, que uno de los principales retos de Maine creaciones ha sido reducir los gastos en la compra de telas.

Debido a que el negocio se ha llevado de una manera informal, las cantidades expresadas en el siguiente trabajo son estimaciones aproximadas del año 2020.

II. Planteamiento del problema

El corte de tela es una actividad importante para la producción de vestidos. El tipo de tela que se utiliza son rollos de tela Tafeta. La dimensión de cada rollo es de $150\,\text{cm} \times 2500\,\text{cm}$. Maine creaciones produce una amplia variedad de diseños de vestidos, pero independientemente del diseño que se realice, las dimensiones de tela a ocupar va a depender únicamente de la talla del vestido. En total Maine produce 11 tallas de vestidos, sin embargo, las tallas más chicas reciben poca demanda, por lo que solo se considerarán las cinco tallas más vendidas, las dimensiones de tela que requiere cada talla se muestran en la tabla (1).

Talla	Dimensiones (en centímetros)
45	33×115
35	26×100
30	24×90
20	16×70
15	13×60

Tabla 1. Tallas de vestidos

Cabe aclarar que debido a las condiciones para la elaboración de los vestidos, el rollo de tela solo se puede cortar a lo largo y no a lo ancho, por lo tanto, los cortes se harán sobre un ancho fijo $W = 150$ cm. Asimismo, los cortes se tomarán sobre la medida más corta de cada talla, es decir, si queremos hacer un corte para la talla 45, de los 150 cm disponibles cortaremos 33 cm. Por lo tanto, un ejemplo de patrón de corte es:

Cortar 3 rollos para la talla 35 y 3 rollos para la talla 30,

justamente este patrón ocupa exactamente los 150 cm de ancho disponible del rollo de tela. En la parte de anexos se muestran los 79 patrones de corte que se tomaron en cuenta.

Para poder saber cuantas unidades son necesarias de cada talla, se revisaron algunas anotaciones que contienen información sobre la venta de vestidos. En la tabla (2) se muestran el número de vestidos vendidos en el año 2020 de cada talla.

Talla	Número de vestidos vendidos
45	150
35	250
30	400
20	350
15	250

Tabla 2. Ventas

A partir de la tabla (2), se puede hacer una aproximación sobre la demanda anual de rollos de cada talla, lo cual se ve reflejado en la tabla(3)

Talla	No. de vestidos en cada orden	No. de unidades requeridas
45	21.7	8
35	25	10
30	27.7	15
20	35.7	10
15	41.6	7

Tabla 3. Órdenes a cubrir

III. Objetivos

III.1. Objetivo general

Crear un modelo que minimice el costo en la inversión de tela.

III.2. Objetivos específicos

- Determinar el número de rollos de tela que Maine creaciones debe utilizar para satisfacer la demanda anual de vestidos.
- Minimizar la cantidad de sobreproducción.
- Minimizar la cantidad de residuos.

IV. Modelo

El modelo que se aplicará es el modelo de corte unidimensional. El número de patrones que se utilizaron fueron 79 , los cuales se muestran explícitamente en el código (1)

Se define $a_{i,j}$ como el número de rollos de ancho i cortado con el patrón j.

A continuación se presenta el modelo matemático que optimiza el número de rollos a utilizar

IV.1. Variables de decisión

La única variable de decisión para nuestro problema es

$$x_j$$

que representa el número de rollos cortados bajo el patrón de corte j donde x_j es un número entero mayor o igual a cero.

IV.2. Restricciones

$$\sum_{j=1}^{79} a_{1,j} x_j \geq 8 \tag{1}$$

$$\sum_{j=1}^{79} a_{2,j} x_j \geq 10 \tag{2}$$

$$\sum_{j=1}^{79} a_{3,j} x_j \geq 15 \tag{3}$$

$$\sum_{j=1}^{79} a_{4,j} x_j \geq 10 \tag{4}$$

$$\sum_{j=1}^{79} a_{5,j} x_j \geq 7 \tag{5}$$

IV.3. Función objetivo:

$$\min z = \sum_{j=1}^{79} x_j \tag{6}$$

V. Resultados

A continuación se muestran los resultados obtenidos al considerar los tres tipos de enfoques.

V.1. Minimizar el número de rollos

El número de rollos mínimos a utilizar para satisfacer la demanda es 8. Mientras que en la sobreproducción es 3 y el residuo es 3. En la tabla (4) se ven reflejados los patrones de corte a utilizar

V.2. Minimizar la cantidad de sobreproducción

El número de rollos mínimos a utilizar para satisfacer la demanda es 8. Mientras que la sobreproducción es 0 y el residuo es 65. En la tabla (5) se muestran los patrones de corte a utilizar.

Talla	Patrón 1	Patrón 2	Patrón 3	Patrón 4
45	2	1	1	1
35	2	2	2	0
30	0	2	0	3
20	2	1	0	2
15	0	0	5	1
Residuos	0	1	0	0
No de cortes	1	3	1	3

Tabla 4. Patrón de cortes para minimizar rollos

Talla	Patrón 1	Patrón 2	Patrón 3	Patrón 4	Patrón 5	Patrón 6
45	2	1	1	1	1	0
35	0	2	2	1	0	3
30	3	1	1	2	1	3
20	0	1	0	2	3	0
15	0	1	3	0	3	0
Residuos	12	12	2	11	6	0
No de cortes	1	1	1	3	1	1

Tabla 5. Patrón de cortes para minimizar sobreproducción

V.3. Minimizar la cantidad de residuos

El número de rollos mínimos a utilizar para satisfacer la demanda es 8. Mientras que la sobreproducción es 6 y el residuo es 0. En la tabla (6) se muestran los patroens de corte a utilizar.

Talla	Patrón 1	Patrón 2	Patrón 3	Patrón 4	Patrón 5	Patrón 6	Patrón 7	Patrón 8
45	2	2	1	1	1	1	1	0
35	2	0	4	2	0	0	0	3
30	0	0	0	0	3	1	0	3
20	2	2	0	0	2	5	0	0
15	0	4	1	5	1	1	9	0
Residuos	0	0	0	0	0	0	0	0
No de cortes	1	1	1	1	1	1	1	1

Tabla 6. Patrón de cortes para minimizar sobreproducción

VI. Conclusiones

En el año 2020 para cubrir la demanda anual se compraron 12 rollos de tela. Por lo tanto, el ahorro que se tendría al implementar el modelo expuesto sería de 4 rollos, lo que equivale a un ahorro de $ 2500 pesos. Un aspecto interesante a notar es que independientemente del enfoque que se elija optimizar, el número mínimo de rollos se mantiene constante. Al exponer las soluciones obtenidas, el enfoque que más le agradó a Maine creaciones fue el de minimizar residuos. Asimismo, se pretende abordar un problema bidimensional, pero, debido a que no se tiene una demanda constante mensual, lo que sería de mayor utilidad sería minimizar el residuo individual de cada rollo y para cubrir la complejidad de corte, de tener una solución factible se procedería a buscar alguna herramienta de molde para facilitar el corte.

VII. Anexos

Código 1. Problema de corte

```
from ortools.linear_solver import pywraplp

###Variables de decision

model = pywraplp.Solver.CreateSolver('SCIP')

# Datos

# Definimos el número de demandas (órdenes) de cada talla
d = [8, 10, 15, 10, 7]

# Definimos la matriz que contiene los patrones de corte la última
# columna representa el residuo de realizar el corte con el patrón

aji=[[4, 0, 0, 0, 1, 5],
      [3, 1, 1, 0, 0, 1],
      [3, 1, 0, 1, 0, 9],
      [3, 1, 0, 0, 1, 12],
      [3, 0, 2, 0, 0, 3],
      [3, 0, 1, 1, 0, 11],
      [3, 0, 1, 0, 2, 1],
      [3, 0, 0, 3, 0, 3],
      [3, 0, 0, 2, 1, 6],
      [3, 0, 0, 1, 2, 9],
      [3, 0, 0, 0, 3, 12],
      [2, 3, 0, 0, 0, 6],
      [2, 2, 1, 0, 0, 8],
      [2, 2, 0, 2, 0, 0],
      [2, 2, 0, 1, 1, 3],
      [2, 2, 0, 0, 2, 6],
      [2, 1, 2, 0, 0, 10],
      [2, 1, 1, 2, 0, 2],
      [2, 1, 1, 1, 1, 5],
      [2, 1, 1, 0, 2, 8],
      [2, 0, 3, 0, 0, 12],
      [2, 0, 2, 2, 0, 4],
      [2, 0, 2, 1, 1, 7],
      [2, 0, 2, 0, 2, 10],
      [2, 0, 1, 3, 0, 12],
      [2, 0, 1, 2, 2, 2],
      [2, 0, 1, 1, 3, 5],
      [2, 0, 1, 0, 4, 8],
      [2, 0, 0, 5, 0, 4],
      [2, 0, 0, 4, 1, 7],
      [2, 0, 0, 3, 2, 10],
      [2, 0, 0, 2, 4, 0],
      [2, 0, 0, 1, 5, 3],
```

```
       [2, 0, 0, 0, 6, 6],
       [1, 4, 0, 0, 1, 0],
       [1, 3, 1, 0, 1, 2],
       [1, 2, 2, 1, 0, 1],
       [1, 2, 1, 2, 0, 9],
       [1, 2, 1, 1, 1, 12],
       [1, 2, 1, 0, 3, 2],
       [1, 2, 0, 4, 0, 1],
       [1, 2, 0, 3, 1, 4],
       [1, 2, 0, 2, 2, 7],
       [1, 2, 0, 1, 3, 10],
       [1, 2, 0, 0, 5, 0],
       [1, 1, 3, 1, 0, 3],
       [1, 1, 3, 0, 1, 6],
       [1, 1, 2, 2, 0, 11],
       [1, 1, 2, 1, 2, 1],
       [1, 1, 2, 0, 3, 4],
       [1, 0, 4, 1, 0, 5],
       [1, 0, 4, 0, 1, 8],
       [1, 0, 3, 2, 1, 0],
       [1, 0, 3, 1, 2, 3],
       [1, 0, 3, 0, 3, 6],
       [1, 0, 2, 4, 0, 5],
       [1, 0, 2, 3, 1, 8],
       [1, 0, 2, 2, 2, 11],
       [1, 0, 2, 1, 4, 1],
       [1, 0, 2, 0, 5, 4],
       [1, 0, 1, 5, 1, 0],
       [1, 0, 1, 4, 2, 3],
       [1, 0, 1, 3, 3, 6],
       [1, 0, 1, 2, 4, 9],
       [1, 0, 1, 1, 5, 12],
       [1, 0, 1, 0, 7, 2],
       [1, 0, 0, 7, 0, 5],
       [1, 0, 0, 6, 1, 8],
       [1, 0, 0, 5, 2, 11],
       [1, 0, 0, 4, 4, 1],
       [1, 0, 0, 3, 5, 4],
       [1, 0, 0, 2, 6, 7],
       [1, 0, 0, 1, 7, 10],
       [1, 0, 0, 0, 9, 0],
       [0, 5, 0, 1, 0, 4],
       [0, 4, 1, 1, 0, 6],
       [0, 4, 0, 2, 1, 1],
       [0, 4, 0, 1, 2, 4],
       [0, 3, 3, 0, 0, 0]]

# Definimos el número de patrones
n = len(aji)
print(n)
# Definimos el número de órdenes
m = len(aji[0])-1

# Variables de decisión
```

```
inf = model.infinity()
X = {}
for j in range(n):
    X[j] = model.IntVar(0,inf,'X{}'.format(j+1))


# Restricciones

for i in range(m):
    model.Add(d[i] <= model.Sum(X[j]*aji[j][i] for j in range(n)))

# Elegimos el tipo de problema que deseamos resolver

# Minimizar el número de rollos
model.Minimize(model.Sum(X[j] for j in range(n)))


# Minimizar los residuos
#model.Minimize(model.Sum(X[j]*aji[j][5] for j in range(n)))

# Minimizar la sobre producción
#model.Minimize(model.Sum(model.Sum(X[j]*aji[j][i] for j in range(n))-d[i]
# for i in range(m)))

status = model.Solve()

if status == pywraplp.Solver.OPTIMAL or status == pywraplp.Solver.FEASIBLE:
        print('Solución:')
        print('Valor de la funcion objetivo =',
                sum(X[j].solution_value() for j in range(n)))
        for j in range(n):
            if X[j].solution_value() != 0:
                print("Para el patron", aji[j])
                print("Número de patrones utilizados =", X[j].solution_value())
        print("Sobre produccion",sum(X[j].solution_value()*aji[j][i]
        for j in range(n) for i in range(m))-sum(d[i] for i in range(m)))
        print("Residuos=",sum(X[j].solution_value()*aji[j][5]
                for j in range(n)))
```