

21 DE JANEIRO DE 2019



**Universidade do Minho**  
Departamento Produção e Sistemas

Mestrado em Engenharia de Sistemas

## TRABALHO PRÁTICO

**Autores:**

Ana Margarida Rolim Marques pg38332

Célia Figueiredo a67637

Daniel Sousa pg37112

Márcia Costa a67672

**Docente:** Cláudio Manuel Martins Alves

# Índice

Índice .....	1
Índice de Figuras .....	2
Índice de Tabelas .....	3
GRUPO I .....	4
1. Introdução à Teoria de Decisão.....	4
GRUPO II .....	7
1. Modelos e Métodos de Decisão .....	7
1.1. Descrição do problema .....	7
1.2. Objetivo .....	8
1.3. Variáveis .....	8
1.4. Restrições do Problema .....	10
1.5. Função Objetivo .....	12
1.6. Análise de Resultados .....	13
2. Anexos .....	17
2.1. Anexo 1 .dat .....	17
2.2. Anexo 2. Script com o modelo incompleto.....	18
2.3. Anexo 3. Script com o modelo completo.....	21

# Índice de Figuras

Figura 1 - Árvore de Probabilidade.....	4
Figura 2 - Representação do esquema da cadeia de abastecimento.....	7

# Índice de Tabelas

Tabela 1: Procura de cada cliente de cada produto.....	13
Tabela 2: Extensão da capacidade.....	14
Tabela 3: Quantidade de produto entregue a cada cliente nos 45 dias .....	15
Tabela 4: Produção nos primeiros 25 dias de cada produto.....	16
Tabela 5: Produção de cada produto do dia 26 ao 45.....	16

# GRUPO I

## 1. Introdução à Teoria de Decisão

Problema:

Três máquinas diferentes são usadas para produzir artigos em lotes.

A máquina A produz 25% dos artigos e 4% destes são defeituosos.

A máquina B produz 30% dos artigos e 2% deles defeituosos.

A máquina C produz os 45% dos artigos, onde 1% deles são defeituosos.

Acontecimentos:

A: Artigo é produzido pela máquina A;

B: Artigo é produzido pela máquina B;

C: Artigo é produzido pela máquina C;

D: Artigo é Defeituoso.

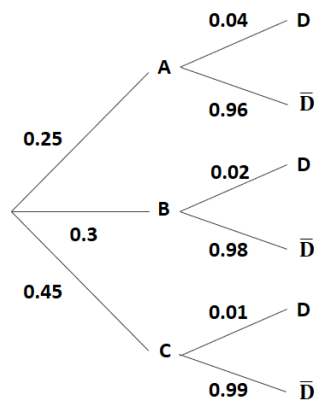


Figura 1 - Árvore de Probabilidade

Todos os acontecimentos são independentes.

Através da Árvore de Probabilidades:

- $P(A \cap B) = 0.25 * 0.04 = 0.01$ , é a probabilidade dos artigos serem produzidos pelas máquinas A e B
- $P(B \cap D) = 0.03 * 0.02 = 0.006$ , é a probabilidade dos artigos serem produzidos pelas máquinas B e C

- $P(C \cap D) = 0.45 * 0.01 = 0.0045$ , é a probabilidade dos artigos serem produzidos pelas máquinas B e C

**1. Qual é a probabilidade de um artigo escolhido aleatoriamente num lote ser defeituoso?**

A probabilidade pretendida é  $P(D)$ .

Assim:

$$P(D) = P(A \cap D) + P(B \cap D) + P(C \cap D) = 0.01 + 0.006 + 0.0045 = 0.0205$$

Conclui-se que a probabilidade de um artigo escolhido aleatoriamente, ser defeituoso, é de 2.05%.

**2. Qual é a probabilidade de um artigo defeituoso, escolhido aleatoriamente, ter sido produzido pela máquina B?**

A probabilidade pretendida é  $P(D/B)$ .

De acordo com a fórmula da probabilidade condicionada, sabemos que

$$P(D/B) = \frac{P(D \cap B)}{P(B)}$$

Sabe-se que  $P(B \cap D) = P(D \cap B)$ .

Assim:

$$P(D/B) = \frac{0.3 * 0.02}{0.3} = 0.02$$

Conclui-se que a probabilidade de um artigo defeituoso, escolhido aleatoriamente, ter sido produzido pela máquina B é de 2%.

**3. Considere uma caixa de 100 artigos produzidos na mesma máquina, e considere ainda que a máquina em questão é desconhecida. Um artigo é escolhido aleatoriamente a partir desse lote, e verifica-se que esse mesmo artigo é defeituoso.**

A partir do exercício 1, verifica-se que 2.05% dos artigos produzidos são defeituosos, o que implica que, em 100 artigos, 2 destes são defeituosos.

a) Qual é a probabilidade do artigo escolhido a seguir ser também ele defeituoso?

Acontecimentos:

P1: "Primeiro artigo é defeituoso"

P2: "Segundo artigo é defeituoso"

Pretende-se a probabilidade de  $P(P2/P1)$ .

Sabe-se que  $P(E) > 0$  e, pela fórmula da probabilidade condicionada, tem-se que:

$$P(P2) = \frac{P(P2 \cap P1)}{P(P1)} = \frac{\frac{2.05 - 1}{99} - \frac{2.05}{100}}{\frac{2.05}{100}} = 0.0106$$

Assim, a probabilidade do segundo artigo escolhido ser defeituoso é 1.06%.

- b) Suponha que o primeiro artigo escolhido não era defeituoso. Nesse caso, qual seria a probabilidade do segundo artigo escolhido ser defeituoso?

Acontecimentos:

$\overline{S1}$ : "O primeiro artigo não é defeituoso"

$P2$ : "O segundo artigo é defeituoso"

Pretende-se a probabilidade de  $P(S2/\overline{S1})$ .

Uma vez que a probabilidade do primeiro artigo não ser defeituoso é

$$P(\overline{S1}) = \frac{100 - 2.05}{100} = 0.9795$$

Assim  $P(P2/\overline{S1})$ , pela probabilidade condicionada é

$$P(P2/\overline{S1}) = \frac{P\left(\frac{P2}{\overline{S1}}\right)}{P(\overline{S1})} = \frac{\frac{2.05 - 1}{99} * \frac{100 - 2.05}{100}}{\frac{100 - 2.05}{100}} = 0.0207$$

Posto isto, a probabilidade do segundo artigo escolhido ser defeituoso, sabendo que o primeiro escolhido não é defeituoso é de 2,07%.

# GRUPO II

## 1. Modelos e Métodos de Decisão

### 1.1. Descrição do problema

A XXS é uma companhia que produz três tipos de produtos: P1, P2 e P3. O P1 é o produto de gama alta, o P2 o de média gama e o P3 o de gama baixa. A XXS fornece três clientes: C1, C2 e C3. Todos os produtos são fabricados na fábrica F1 que tem uma única linha de produção R1. Os três produtos competem assim pela capacidade instalada em R1. Para fabricar os produtos, são necessários dois componentes X1 e X2. O fornecedor principal da XXS é designado por S1, e pode fornecer ambos os componentes X1 e X2. A XXS recorre também quando necessário a um fornecedor secundário (S2) que fornece apenas o componente X1, mas a um custo superior. A Figura 2 ilustra a estrutura da cadeia de abastecimento dessa companhia.

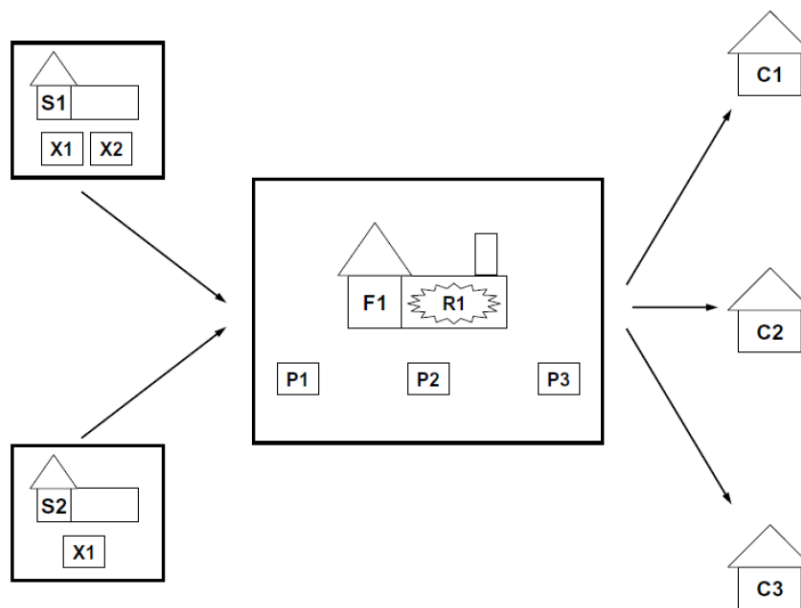


Figura 2 - Representação do esquema da cadeia de abastecimento

A procura efetiva dos clientes e as previsões da procura futura determinam o processo de planeamento. Sempre que um cliente não é servido no prazo máximo de 30 dias, a venda é considerada como perdida e a penalidade associada a uma “não entrega” é aplicada. As restrições “fortes” do problema são a capacidade de produção em R1 e a quantidade máxima de componentes X1 que S1 pode fornecer. Neste último caso, o limite é modelado como uma restrição à capacidade de transporte T1. No caso de um aumento não previsto da procura, a capacidade de R1 pode ser aumentada até ao dobro com um determinado custo associado. Se a capacidade de transporte T1 estiver no seu limite, o fornecimento de matérias-primas terá de ocorrer mais cedo, ou em alternativa,



a XXS poderá recorrer ao fornecedor S2 por um preço de compra mais elevado. O fabrico de cada um dos produtos requer uma determinada quantidade de cada um dos componentes X1 e X2. Em função da gama do produto, o processo de produção em R1 é mais ou menos complicado, o que se traduz em diferentes níveis de consumo para cada um dos produtos P1, P2 e P3.

## 1.2. Objetivo

O problema consiste em determinar o plano de produção mais económico que garanta os níveis de inventário mais baixos e a maior satisfação dos clientes atendendo às seguintes parcelas de custo: produção; extensão da capacidade produtiva; fornecimento externo; armazenamento; entregas em atraso e falhas nas entregas.

## 1.3. Variáveis

- **Índices**

$i$ : produto  $i$

$j$ : cliente  $j$

$c$ : componente  $c$

$s$ : fornecedor  $s$

- **Relacionadas com os custos:**

$P_{id}$  : Procura do produto  $i$  no dia  $d$

$CP_i$  : Custo de produção do produto  $i$

$CA_{id}$ : Custo por Atraso do produto  $i$  no dia  $d$

$CNE_i$ : Custo por falha de entrega do produto  $i$

$CARM_i$ : Custo de armazenamento do produto  $i$  no dia  $d$

$CARM_{cd}$ : Custo de armazenamento do componente  $c$  no dia  $d$

$CF_{cs}$ : Custo do componente  $c$ , fornecido pelo fornecedor  $s$

$CEC$ : Custo de extensão da capacidade

- **Variáveis de Decisão**

$EC$ : Extensão de capacidade (hora)

$Q_{id}$ : Quantidade produzida do produto  $i$  por dia  $d$

$QA_{id}$ : Quantidade do produto  $i$ , atrasado por dia  $d$

$QNE_i$ : Quantidade não entregue do produto  $i$

$QARM_i$ : Quantidade armazenada do produto  $i$

$QARM_c$ : Quantidade armazenada do componente  $c$

$Q_c$ : Quantidade do componente  $c$

$Q_{cs}$ : Quantidade do componente  $c$  fornecida pelo fornecedor  $s$

$QE_{ij}$ : Quantidade entregue do produto  $i$  ao cliente  $j$

$NE_{ijd} : \begin{cases} 1, & \text{se há uma não entrega do produto } i \text{ ao cliente } j \text{ no dia } d \\ 0, & \text{caso contrário} \end{cases}$

$A_{ijd} : \begin{cases} 1, & \text{se há um atraso do produto } i \text{ ao cliente } j \text{ no dia } d \\ 0, & \text{caso contrário} \end{cases}$

$EBin_d : \begin{cases} 1, & \text{se há uma extensão da capacidade no dia } d \\ 0, & \text{caso contrário} \end{cases}$

- **Variáveis**

$CR_s$ : Capacidade de resposta do fornecedor  $s$

$T_{cs}$ : Capacidade de transporte do fornecedor  $s$  para o componente  $c$

$LT$ : prazo máximo de entrega

$DP_i$ : Dia pretendido para entrega do produto  $i$  ao cliente

$DE_i$ : Dia em que o produto  $i$  é entregue ao cliente

$DA$ : Dias de atraso

- **Parâmetros**

$CR1_i$ : Consumo de R1, em horas, por produto  $i$

$C_{ci}$ : Consumo do componente  $c$  por produto  $i$

$TX_i$ : Taxa de produção do produto  $i$

## 1.4. Restrições do Problema

### Restrição 1: Capacidade de Produção

$$\text{Fórmula Geral: } \sum_{d=1}^{45} \sum_{i=1}^3 Q_{id} * \frac{CR1_i}{TX_i} \leq \text{HorasDia} + EC_d * EBin_d \quad (1)$$

Com esta restrição garantiu-se que a quantidade produzida por dia de todos os produtos não excedia a capacidade de produção da linha, em horas. Como a capacidade da linha podia ser estendida, foi necessário somar a variável  $EC$  ao tempo normal de funcionamento da linha, por dia.

### Restrição 2: Procura

$$\text{Forma geral: } \sum_{d=1}^{45} \sum_{j=1}^3 \sum_{i=1}^3 (QE_{ijd} + QARM_{id}) \geq P_{id} \quad (2)$$

A restrição 2 garante que a procura do cliente pelo produto  $i$  é satisfeita. A quantidade de produto enviada para cada cliente e a quantidade desse mesmo produto que está armazenada, terá de ser superior à procura desse mesmo produto.

### Restrição 3: Oferta/Capacidade de Resposta do Fornecedor

$$\text{Forma Geral: } \sum_{d=1}^{45} \sum_{c=1}^2 \sum_{s=1}^2 Q_{dcs} \leq \sum_{d=1}^{45} \sum_{c=1}^2 \sum_{s=1}^2 CR_{dcs} \quad (3)$$

A terceira restrição impede, o fornecedor, de enviar produtos que não tenha em stock. Isto é, a quantidade de componentes que o fornecedor pode enviar para a fábrica, tem de ser menor ou igual à quantidade desse mesmo componente, em stock.

### Restrição 4: Capacidade de Transporte

Forma Geral:  $Q_{11} \leq T_{11}$  (4)

A restrição 4, garante que o fornecedor S, só poderá enviar produtos para a fábrica, tendo em consideração a capacidade de transporte destes. Isto implica que a quantidade fornecida de produto 1, será restringida pela capacidade de transporte desse mesmo produto.

#### **Restrição 5: Armazenagem**

Forma geral:  $QARM_{id} + Q_{id} \geq QE_{ij}$  (5)

A quinta restrição impede, a fábrica, de enviar produtos ao cliente, que não tenha em stock. Isto é, a quantidade de componentes que o fornecedor pode enviar para o cliente, tem de ser menor ou igual à quantidade desse mesmo componente, em stock e à quantidade produzida desse mesmo produto.

#### **Restrição 6: Entrega**

Considera-se que uma falha de entrega sempre que o dia de entrega excede o prazo de 30 dias.

#### **Restrição 7: Atraso**

Forma Geral:  $DA_i = DE_i - DP_i$  (6)

A restrição 7, implica que, dados o dia pretendido para a entrega de uma encomenda, ao cliente, e o dia em que essa mesmo foi entregue, seja calculado o número de dias de atraso da encomenda, caso se verifique atraso.

#### **Restrição 8:**

Forma Geral:  $QE_{ijd} \leq Q_{id-1} - QARM_{id-1}$  (7)

A restrição 8, implica que, a quantidade de produtos i entregues ao cliente j, tenham de ser produzidos e armazenados no dia anterior.

### Restrição 9: Quantidade entregue

$$\text{Forma Geral: } \sum_{d=1}^{45} \sum_{i=1}^3 QE_{id} \leq P_{id} \quad (8)$$

Só se entrega, no máximo, a quantidade procurada.

### Restrição 10: Componentes

$$\text{Forma Geral: } \sum_{c=1}^2 \sum_{i=1}^3 C_{ci} \leq TX_i \quad (9)$$

Os componentes  $c$  necessários para produzir cada produto  $i$ , terá de ser menos que a taxa de produção do produto  $i$ .

## 1.5. Função Objetivo

O objetivo do problema, como já mencionado, passa por determinar o plano de produção mais económico que garanta níveis de inventário mais baixos e uma maior satisfação dos clientes, atendendo aos custos de produção, aos custos de extensão de capacidade produtiva, ao custo do fornecimento externo, ao custo de armazenamento, ao custo de entregas atrasadas e ao custo das falhas de entrega.

Posto isto, a forma geral da função objetivo é a seguinte:

$$\begin{aligned} FO: \text{Min } Z = & \sum_{d=1}^{45} \sum_{i=1}^3 Q_{id} * CP_i \\ & + \sum_{d=1}^{45} \sum_{c=1}^2 \sum_{s=1}^2 Q_{csd} * CF_{cs} + \sum_{d=1}^{45} \sum_{i=1}^3 QARM_{id} * CARM_i \\ & + \sum_{d=1}^{45} \sum_{c=1}^2 QARM_c * CARM_{cd} + \sum_{d=1}^{45} \sum_{i=1}^3 \sum_{j=1}^3 QA_{id} * CA_{id} * A_{ijd} \\ & + \sum_{d=1}^{45} \sum_{j=1}^3 \sum_{i=1}^3 QNE_i * CNE_i * NE_{ijd} + \sum_{d=1}^{45} EC_d * CEC * EBin_d \end{aligned} \quad (10)$$

## 1.6. Análise de Resultados

Para a resolução do problema de otimização foram criados dois modelos em linguagem OPL.

O primeiro, engloba todas as restrições supracitadas. No entanto, não foram obtidas soluções.

No segundo, foram retiradas algumas restrições e, os resultados expostos são relativos a esta solução.

Os dois ficheiros estão em anexo, sendo o anexo 3, relativo ao modelo com todas as restrições e, o anexo 2 ao modelo com apenas algumas restrições a partir do qual obtivemos resultados.

Predefiniu-se a procura de cada cliente para cada produto, no espaço temporal de 45 dias.

Estes dados estão expostos na tabela 1.

*Tabela 1: Procura de cada cliente de cada produto*

Período	Cliente 1			Cliente 2			Cliente 3		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
1	2000	200	1000	200	200	100	0	0	0
2	300	300	0	0	0	0	100	100	1000
3	200	200	100	0	0	100	0	0	0
4	100	100	100	0	1000	0	0	100	0
5	1000	1000	100	100	0	0	100	0	100
6	0	0	0	0	0	100	0	0	0
7	0	0	0	100	100	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	200	200	100	0	0	0	100	100	100
10	0	0	0	0	0	0	0	0	0
11	100	100	100	100	100	100	0	0	0
12	0	0	0	0	0	0	0	100	0
13	0	0	100	0	0	0	0	0	100
14	0	0	0	100	100	0	0	0	0
15	2000	200	0	0	0	100	0	0	0
16	0	0	100	0	0	0	0	0	0
17	0	0	0	0	0	0	300	300	100
18	0	0	0	200	200	100	0	0	0
19	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	100	0	0	0
21	0	0	0	0	0	0	0	0	0
22	400	400	100	0	0	0	400	400	100
23	0	0	0	300	300	100	0	0	0
24	0	0	0	0	0	0	0	2000	0
25	0	0	0	0	0	0	0	0	0
26	100	200	0	0	100	0	0	0	100
27	0	0	3000	0	0	0	200	0	0

28	0	0	0	2000	0	0	0	100	0
29	2000	0	0	0	100	100	0	0	0
30	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	100	1000
32	0	0	0	0	0	0	0	0	0
33	0	0	0	800	800	100	100	0	0
34	100	100	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0
36	0	0	1000	0	100	0	0	100	2000
37	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	100	2000	0	0
39	0	100	0	100	0	0	0	0	0
40	100	0	0	0	0	0	0	0	0
41	0	0	100	0	0	0	0	0	0
42	0	0	0	10	100	0	500	500	100
43	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0

O modelo OPL do anexo 2, gerou os seguintes resultados:

- Extensão de capacidade

Tabela 2: Extensão da capacidade

periodo	Value	periodo	Value
1	6	26	0
2	0	27	4
3	0	28	4
4	0	29	4
5	0	30	0
6	0	31	0
7	0	32	0
8	0	33	0
9	0	34	0
10	0	35	0
11	0	36	4
12	0	37	0
13	0	38	4
14	0	39	0
15	4	40	0
16	0	41	0
17	0	42	0
18	0	43	0
19	0	44	0
20	0	45	0
21	0		
22	0		
23	0		
24	0		
25	0		





- Quantidade produzida, de cada produto, em 45 dias

*Tabela 4: Produção nos primeiros 25 dias de cada produto*

Período	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
P1	6600	2200	400	200	100	1200	0	100	0	300	0	200	0	0	100	2000	0	300	200	0	0	0	800	300	0	0
P2	1200	400	400	200	1200	1000	0	100	0	300	0	200	100	0	100	200	0	300	200	0	0	0	800	300	2000	0
P3	3300	1100	1000	200	100	200	100	0	0	200	0	200	0	200	0	100	100	100	100	0	100	0	200	100	0	0

*Tabela 5: Produção de cada produto do dia 26 ao 45*

Período	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
P1	100	200	2000	2000	0	0	0	900	100	0	0	0	2000	100	100	0	510	0	0	0
P2	300	0	100	100	0	100	0	800	100	0	200	0	0	100	0	0	600	0	0	0
P3	100	3000	0	100	0	1000	0	100	0	0	3000	0	100	0	0	100	100	0	0	0

Assim, o valor da função objetivo obtido é 816610.

## 2. Anexos

### 2.1. Anexo 1 .dat

```
/******  
* OPL 12.8.0.0 Data  
* Creation Date: 19/01/2019 at 15:16:06  
*****/  
Produtos={"P1","P2","P3"};  
Clientes={"C1","C2","C3"};  
Componentes={"X1","X2"};  
Fornecedores={"S1","S2"};  
  
//periodo de estudo  
P=45;  
  
//capacidade de R1  
capacidadeR1 = 8;  
  
capacidadeTransporte=1000;  
  
//custo da Extensao da capacidade  
custoExtensaoCapacidade=2000;  
  
//Custo de produzir cada Produto  
custoProducao = [25,15,15];  
  
//custos pelos Atrasos  
custoAtraso= [10,5,5];  
  
//custos dos atrasos  
custoNaoEntrega= [350,250,180];  
  
//custo de armazenamento dos tres produtos  
custoArmazenamentoProduto = [2,1,1];  
  
//Custo de armazenamento de uma componente  
custoArmazenarComponente = [0.5 ,0.5];  
  
/*  
//Custo das componentes  
custoUnitarioPorComponenteS1= [60, 30]; //componente X1  
custoUnitarioPorComponenteS2= [80, 0];  
*/  
  
custoUnitarioPorComponente= [[60, 30],[80,0]];  
  
//consumo da procura de X1 e X2  
consumoComponentes = [[3000,1000],[2000,1000],[1000,1000]];
```

```

//Consumo em horas de R1
consumoR1 = [6,4,4];

//Tempo de processamento
tempoDeProcessamentoProdutos = [8,8,8];

//Taxa de producao
taxaProducao = [1000, 1000, 1000];

prazo = 30;

custoExtraTransporte=500;

//int procura[Produtos][Clientes][periodo]
procura = [
[
[200,300,200,100,1000,0,0,0,200,0,100,0,0,0,2000,0,0,0,0,0,400,0,0,0,100,0,0,2000,0,
0,0,0,100,0,0,0,0,0,100,0,0,0,0,0],
[200,0,0,0,100,0,100,0,0,0,100,0,0,100,0,0,0,200,0,0,0,0,300,0,0,0,0,2000,0,0,0,0,800,0,
0,0,0,0,100,0,0,10,0,0,0,0],
[0,100,0,0,100,0,0,0,0,100,0,0,0,0,0,0,0,300,0,0,0,0,400,0,0,0,0,200,0,0,0,0,0,100,0,0,0,0,
2000,0,0,0,500,0,0,0]],
[
[200,300,200,100,1000,0,0,0,200,0,100,0,0,0,200,0,0,0,0,0,400,0,0,0,200,0,0,0,0,0,0,
0,100,0,0,0,0,100,0,0,0,0,0,0],
[200,0,0,1000,0,0,100,0,0,0,100,0,0,100,0,0,0,200,0,0,0,0,300,0,0,100,0,0,100,0,0,0,80
0,0,0,100,0,0,0,0,0,100,0,0,0],
[0,100,0,100,0,0,0,0,100,0,0,100,0,0,0,0,300,0,0,0,0,400,0,2000,0,0,0,100,0,0,100,0,0,0,
0,100,0,0,0,0,500,0,0,0]],
[
[1000,0,100,100,100,0,0,0,100,0,100,0,100,0,0,100,0,0,0,0,100,0,0,0,0,3000,0,0,0,0,0,
0,0,0,1000,0,0,0,0,100,0,0,0,0],
[100,0,100,0,0,100,0,0,0,0,100,0,0,100,0,0,100,0,100,0,0,100,0,0,0,0,100,0,0,0,100,
0,0,0,0,100,0,0,0,0,0,0,0],
[0,1000,0,0,100,0,0,0,100,0,0,0,100,0,0,0,100,0,0,0,0,100,0,0,0,0,1000,0,0,0,
0,2000,0,0,0,0,0,100,0,0,0]]];

```

## 2.2. Anexo 2. Script com o modelo incompleto

```

/*****
* OPL 12.8.0.0 Model

```

```

* Creation Date: 19/01/2019 at 15:15:40
*****/
{string} Produtos =...;
{string} Componentes =...;
{string} Fornecedores=...;
{string} Clientes = ...;

int P=...;

range periodo = 1..P;
range periodoArmazenamento = 1..P+1;

//Capacidade
int capacidadeR1=...;
//extensao da capacidade horas de trabalho no maximo 16

int taxaProducao[Produtos]=...;

int custoAtraso[Produtos]=...;

int custoNaoEntrega [Produtos]=...;

int tempoDeProcessamentoProdutos[Produtos]=...;

int prazo=...;

//capacidade de transporte T1
int capacidadeTransporte=...;

int custoExtraTransporte=...;

/*Parametros*/

int custoExtensaoCapacidade=...;

int custoProducao[Produtos]=...;

int consumoComponentes[Produtos][Componentes]=...;

int consumoR1[Produtos]=...;

int custoArmazenamentoProduto[Produtos]=...;

float custoArmazenarComponente[Componentes]=...;

int custoUnitarioPorComponente[Fornecedores][Componentes]=...;

int procura[Produtos][Clientes][periodo]=...;

/* Variaveis Decisao */
//Quantidade do produto p produzido no periodo i
dvar int+ QuantidadeProduzida [Produtos][0..P] ;

//extensao capacidade
dvar int+ extensaoCapacidade [periodo] ;

//Quantidade entregue do produto i no dia j ao cliente x
dvar int+ quantidadeEntregue[Produtos][Clientes][periodo];

```

```

//quantidades dos componentes produzidos
dvar int+ quantidadeComponentesConsumidos[Componentes][periodo];

//Quantidade Armazenada dos produtos
dvar int+ quantidadeArmazenadaP[Produtos][periodoArmazenamento];

//Quantidade Armazenada dos componentes
dvar int+ quantidadeArmazenadaC[Componentes][periodoArmazenamento];

//    Quantidade de produto em atraso
dvar int+ quantidadeAtraso[Produtos][Clientes][periodo];

//Quantidade de artigos não entregues
dvar int+ quantidadeNaoEntregue[Produtos][periodo];

//Quantidade enviada para os fornecedores
dvar int+ ofertaFornecedores[Fornecedores][Componentes][periodo];

/*Funcao Objetivo*/
minimize sum ( i in periodo)
(
    sum (p in Produtos) QuantidadeProduzida[p][i]*custoProducao[p]+
    sum (f in Fornecedores, c in Componentes)
ofertaFornecedores[f][c][i]*custoUnitarioPorComponente[f][c]+
    sum (p in Produtos, c in Clientes)
quantidadeAtraso[p][c][i]*custoAtraso[p]+ //custos do atraso
    sum (p in Produtos) quantidadeNaoEntregue[p][i]*custoNaoEntrega[p]+
    custoExtensaoCapacidade*extensaoCapacidade[i] + //custo das horas
extras
    sum (p in Produtos)
custoArmazenamentoProduto[p]*quantidadeArmazenadaP[p][i] + //custo de
armazenamento
    sum (comp in Componentes)
quantidadeArmazenadaC[comp][i]*custoArmazenarComponente[comp]
);

/*Restricoes*/

subject to {
//1.capacidade de producao, retrição das horas
forall(p in Produtos, i in periodo)

QuantidadeProduzida[p][i]*(consumoR1[p]/taxaProducao[p])<=capacidadeR1+extensaoCapacidade[i];

//limite das horas extras atã no máximo 8
forall (j in periodo)
    extensaoCapacidade[j]<=8;

//2.procura
forall(p in Produtos,d in periodo, c in Clientes)
    quantidadeEntregue[p][c][d] + quantidadeArmazenadaP[p][d] >= sum (c in Clientes) procura[p][c][d];

```

```

        //10.Quantidade Entregue dos produtos serÃ; menor ou igual
        Ã quantidade produzida + a quantidade armazenada do produto
    forall (j in periodo, i in Produtos, m in Clientes)
        quantidadeEntregue[i][m][j] <= QuantidadeProduzida[i][j-1] +
        quantidadeArmazenadaP[i][j];

    //9. Quantidade produzida deverÃ; ser maior ou igual Ã procura dos clientes
    forall (i in periodo, p in Produtos)
        QuantidadeProduzida[p][i]>= sum (c in Clientes)procura[p][c][i];

}

```

## 2.3. Anexo 3. Script com o modelo completo

```

/*****
* OPL 12.8.0.0 Model
* Creation Date: 19/01/2019 at 15:15:40
*****/
{string} Produtos =...;
{string} Componentes =...;
{string} Fornecedores=...;
{string} Clientes = ...;

int P=...;

range periodo = 1..P;
range periodoArmazenamento = 1..P+1;

//Capacidade
int capacidadeR1=...;
//extensao da capacidade horas de trabalho no maximo 16

int taxaProducao[Produtos]=...;

int custoAtraso[Produtos]=...;

int custoNaoEntrega [Produtos]=...;

int tempoDeProcessamentoProdutos[Produtos]=...;

int prazo=...;

//capacidade de transporte T1
int capacidadeTransporte=...;

int custoExtraTransporte=...;

/*Parametros*/

```

```

int custoExtensaoCapacidade=...;

int custoProducao[Produtos]=...;

int consumoComponentes[Produtos][Componentes]=...;

int consumoR1[Produtos]=...;

int custoArmazenamentoProduto[Produtos]=...;

float custoArmazenarComponente[Componentes]=...;

int custoUnitarioPorComponente[Fornecedores][Componentes]=...;

int procura[Produtos][Clientes][periodo]=...;


/* Variaveis Decisao */
//Quantidade do produto p produzido no periodo i
dvar int+ QuantidadeProduzida [Produtos][periodo] ;

//extensão capacidade
dvar int+ extensaoCapacidade [periodo] ;

//Quantidade entregue do produto i no dia j ao cliente x
dvar int+ quantidadeEntregue[Produtos][Clientes][periodo];

//quantidades dos componentes produzidos
dvar int+ quantidadeComponentesConsumidos[Componentes][periodo];

//Quantidade Armazenada dos produtos
dvar int+ quantidadeArmazenadaP[Produtos][periodoArmazenamento];

//Quantidade Armazenada dos componentes
dvar int+ quantidadeArmazenadaC[Componentes][periodoArmazenamento];

//      Quantidade de produto em atraso
dvar int+ quantidadeAtraso[Produtos][Clientes][periodo];

//Quantidade de artigos não entregues
dvar int+ quantidadeNaoEntregue[Produtos][periodo];

//Quantidade enviada para os fornecedores
dvar int+ ofertaFornecedores[Fornecedores][Componentes][periodo];


/*Funcao Objetivo*/

minimize sum ( i in periodo)
(
    sum (p in Produtos) QuantidadeProduzida[p][i]*custoProducao[p]+

```

```

        sum (p in Produtos) quantidadeArmazenadaP[p][i]*custoArmazenamentoProduto[p]+
        sum (f in Fornecedores, c in Componentes)
ofertaFornecedores[f][c][i]*custoUnitarioPorComponente[f][c]+
        sum (p in Produtos, c in Clientes) quantidadeAtraso[p][c][i]*custoAtraso[p]+ //custos
do atraso
        sum (p in Produtos) quantidadeNaoEntregue[p][i]*custoNaoEntrega[p]+
        custoExtensaoCapacidade*extensaoCapacidade[i] + //custo das horas extras
        sum (p in Produtos) custoArmazenamentoProduto[p]*quantidadeArmazenadaP[p][i] +
//custo de armazenamento
        sum (comp in Componentes)
quantidadeArmazenadaC[comp][i]*custoArmazenarComponente[comp]
);

```

/\*Restricoes\*/

```

subject to {
//1.capacidade de producao
forall(p in Produtos, i in periodo)

```

```

QuantidadeProduzida[p][i]*(consumoR1[p]/taxaProducao[p])<=capacidadeR1+extensaoCapaci
dade[i];

```

//2.procura

```

forall(p in Produtos,d in periodo, c in Clientes)
    QuantidadeProduzida[p][d] + quantidadeArmazenadaP[p][d] >= sum (c in Clientes)
    procura[p][c][d];

```

//3.Quantidade de componentes de x1 vindos de s1 e s2

```

forall (f in Fornecedores, i in periodo)
    quantidadeComponentesConsumidos["X1"][i]<= sum (c in Componentes)
ofertaFornecedores[f][c][i];

```

// Quantidade de componentes de x2 vindos de s1 (inutil talvez)

```

forall (i in periodo)
    quantidadeComponentesConsumidos["X2"][i]<=ofertaFornecedores["S1"]["X2"][i];

```

//4.Quantidade entregue dos produtos no periodo j, do cliente m é menor ou igual à procura no produto i, cliente m, no periodo j

```

forall (i in Produtos, m in Clientes)
    sum (j in periodo) quantidadeEntregue[i][m][j] <= sum(j in periodo) procura [i][m][j];

```

//5.Quantidade Entregue dos produtos será menor ou igual à quantidade produzida + a quantidade armazenada do produto

```

forall (j in periodo, i in Produtos)
    sum (m in Clientes) quantidadeEntregue[i][m][j] <= QuantidadeProduzida[i][j] +
    quantidadeArmazenadaP[i][j];

```

//a quantidade armazenada no dia j + a quantidade produzida no dia j - o que entreguei no dia j = quantidade armazenada no dia seguinte



```

    forall (p in Produtos, j in periodo)
quantidadeArmazenadaP[p][j] + QuantidadeProduzida[p][j] - sum (m in Clientes)
quantidadeEntregue[p][m][j] == quantidadeArmazenadaP[p][j+1];

//quantidade de produtos em atraso
forall (p in Produtos, c in Clientes, j in periodo)
    quantidadeAtraso[p][c][j]==procura[p][c][j]-quantidadeEntregue[p][c][j];

/*
//quando adiciono esta restricao deixa de aparecer valores nas quantidades produzidas
//Quantidade de componentes consumidos
forall (j in periodo, c in Componentes)
    quantidadeComponentesConsumidos[c][j] == sum (p in Produtos)
consumoComponentes[p][c]*QuantidadeProduzida[p][j];
*/

//Quantidade armazenada dos componentes
forall (c in Componentes, j in periodo, f in Fornecedores)
    quantidadeArmazenadaC[c][j+1]==quantidadeArmazenadaC[c][j]+ofertaFornecedores[f][c][j]-
quantidadeComponentesConsumidos[c][j];

/*-----*/

// Dias de atraso

//capacidade de transporte T1 para o componente x1 do fornecedor S1
forall (j in periodo)
    ofertaFornecedores["S1"]["X1"][j]<=capacidadeTransporte;

}

```