



**Universidade Federal da Paraíba**  
**Centro de Informática**

**Relatório Final**  
**Problema Mestre de Produção**

Elaine Cristina Lins de Souza;  
Letícia Sousa Leite;  
Stênio Ellison Ferreira Pereira.

Professor: Teobaldo Leite Bulhões Junior.  
Disciplina: Pesquisa Operacional.

JOÃO PESSOA, PB.  
Julho, 2021.

ELAINE CRISTINA LINS DE SOUZA,  
LETÍCIA SOUZA LEITE,  
STÊNIO ELLISON FERREIRA PEREIRA.

## **Relatório Final de Pesquisa Operacional**

Relatório de atividade prática da disciplina de Pesquisa Operacional, do curso de Engenharia de Computação, da Universidade Federal da Paraíba como requisito para a obtenção da terceira nota da disciplina.

Orientador: Teobaldo Leite Bulhões Junior.

**João Pessoa - PB.**

**Julho, 2021.**

## **Resumo**

Este trabalho apresenta um modelo de programação linear, para a obtenção do Plano de Mestre de Produção(PMP) em uma fábrica que produz um determinado produto p1. O objetivo deste trabalho portanto, é colocar em prática o conteúdo aprendido em sala de aula, de modo a modelar e criar um código que funcione para qualquer conjunto de dados e ofereça a resposta de quanto vai ser produzido ou comprado de cada produto que a fábrica necessita em cada semana, e qual o estoque de cada produto ao final de cada semana e qual é o custo desse estoque.

## **Sumário**

<b>Introdução</b>	<b>4</b>
<b>Definição do Problema</b>	<b>4</b>
<b>Modelagem</b>	<b>5</b>
4.1 Dados do Modelo	5
4.2 Definição de Variáveis do Modelo	5
4.3 Função Objetivo e Restrições	6
<b>Implementação</b>	<b>7</b>
<b>Considerações Finais</b>	<b>11</b>
<b>Referências Bibliográficas</b>	<b>12</b>

## 1. Introdução

O capitalismo do mundo atual afeta diretamente o modo em que vivemos. Desde as mudanças nas formas de consumo de bens materiais, até mesmo, em como as organizações produzem esses bens. Dessa forma, as organizações passaram a ter necessidade de inovar suas ideias de produtos, bem como a otimizar o tempo de produção de forma a gerar menos custo e aumentar ainda mais seus lucros para garantir sua competitividade no mercado.

No que diz respeito aos processos produtivos, as organizações devem planejar suas atividades de forma a reduzir seus custos de produção, garantir uma maior produtividade, gerenciar seus produtos de maneira eficaz, e assim fazer o uso de ferramentas que possam auxiliar na tomada de decisões para esses fins. Nesse contexto, o Plano de Mestre de Produção(PMP) que é um documento que lista tudo que será produzido, a quantia e quando a produção acontecerá, torna-se crucial para o sistema de planejamento da produção de uma organização (LEÃO, 2021).

O PMP pode ser dividido em três níveis: i) Planejamento Estratégico; ii) Planejamento Tático; iii) Planejamento Operacional ou Programação da Produção. No nível do Planejamento Operacional ou Programação da Produção as preocupações estão diretamente associadas com o gerenciamento detalhado do dia-a-dia da Fábrica, ou seja, do chamado sequenciamento da produção. Entretanto, essa segmentação no planejamento está inserida em um contexto maior que se configura a partir da abordagem de custos percebida pela empresa.(Souza, 1995 apud JÚNIOR, RODRIGUES E PIZZATO, 2001.)

Diante de tudo isso, nosso trabalho tem como objetivo analisar o controle de gastos de uma determinada fábrica na compra e estocagem de materiais externos para a produção do seu produto, no qual cada um tem seu respectivo preço. Além disso, devemos implementar uma solução, que deve conter um meio de minar estas despesas de produção e armazenamento garantindo uma economia para a fábrica, e colocando em prática assim o PMP.

## 2. Definição do Problema

É implementar um algoritmo de PMP, que funciona do seguinte modo:

Uma fábrica produz o produto final  $p1$ . A demanda por  $p1$  para as próximas  $T$  semanas é conhecida e dada por  $d_t$ ,  $t=1, \dots, T$ . A fábrica dispõe de 800 horas semanais de mão-de-obra. A produção de 1 unidade de  $p1$  exige 2 unidades do produto intermediário  $p2$  e 3 do produto  $c1$ . Por sua vez, a produção de 1 unidade do produto  $p2$  exige 1 unidade do produto  $c1$  e 2 do produto  $c2$ . Os produtos  $p1$  e  $p2$  são produzidos na própria fábrica, cada unidade de  $p1$  ou  $p2$  produzida consome 1 hora de mão-de-obra. Os produtos  $c1$  e  $c2$  são comprados externamente.

Os custos de aquisição são de  $cc1$  e  $cc2$  reais por unidade. Entretanto, cada pedido de compra (que pode ser dos dois produtos) tem um custo fixo de  $CF$  reais. Todos os 4 tipos de produtos podem ser mantidos em estoque de uma semana para outra, entretanto existe um custo de  $ep1$ ,  $ep2$ ,  $ec1$ ,  $ec2$  reais por unidade de estoque.

O objetivo é decidir para as  $T$  semanas o quanto vai ser produzido de cada produto  $p1$  e  $p2$  e quanto se vai comprar de  $c1$  e de  $c2$  de forma a atender todas as demandas e minimizar o custo total com compras e estoques. Assuma que o estoque inicial de todos os produtos é zero.

### 3. Modelagem

#### 4.1 Dados do Modelo

$T$  = quantidade de semanas;

$CC_i$  = custo de compra por unidade do insumo  $i$ ;

$DP_{it}$  = demanda do produto  $i$  no período  $t$ ;

$VP_i$  = custo de estoque por unidade do produto  $i$  por semana;

$VC_i$  = custo de estoque por unidade do insumo  $i$  por semana;

$CFP$  = custo fixo por pedido

$H_i$  = horas de produção por unidade;

$HT$  = Horas máximas de trabalho por semana ( $HT = 800$ )

#### 4.2 Definição de Variáveis do Modelo

$P_{it}$  = quantidade a ser produzida do produto  $i$  no período  $t$  ( $i = \{1,2\}$ ,  $t = 1, \dots, T$ );

$C_{it}$  = quantidade a ser comprada do insumo  $i$  no período  $t$  ( $i = \{1,2\}$ ,  $t = 1, \dots, T$ );

$EP_{it}$  = estoque do produto  $i$  ao final do período  $t$  ( $i = \{1,2\}$ ;  $t = 1, \dots, T$ );

$EC_{jt}$  = estoque do insumo  $i$  ao final do período  $t$  ( $i = \{1,2\}$ ;  $t = 1, \dots, T$ );

**CFt** = indica se foi realizado pedido de insumos ou não;

#### 4.3 Função Objetivo e Restrições

$$\text{Min} \sum_{j=1}^n \sum_{t=1}^T ((Cjt \times CCj) + (EPjt \times VPj) + (ECjt \times VCj))$$

Essa função tem como objetivo minimizar para as T semanas o valor gasto nas compras dos insumos c1 e de c2 , somados ao preço de estocagem por semana dos produtos p1, p2, c1 e c2, de forma a atender todas as demandas.

**Sujeito a:**

$$EP1t = P1t - DP1t \quad t = 1, \dots, T \quad (\text{na primeira semana})$$

$$EP1t = EP1(t-1) + P1t - DP1t \quad t = 1, \dots, T \quad (1)$$

(Estoque do produto P1 atual = Estoque do produto P1 da semana anterior + Quant. produzida de P1 na semana atual - Demanda de P1 na semana atual)

$$EP2t = P2t - (DP1t \times 2) \quad t = 1, \dots, T \quad (2) \quad (\text{na primeira semana})$$

$$EP2t = EP2(t-1) + P2t - (DP1t \times 2) \quad t = 1, \dots, T \quad (2)$$

(Estoque do produto P2 atual = Estoque do produto P2 da semana anterior + Quant. produzida de P2 na semana atual - (Demanda de P1 na semana atual x 2))

$$EC1t = C1t - (P2t) - (DP1t \times 3) \quad t = 1, \dots, T \quad (\text{na primeira semana})$$

$$EC1t = EC1(t-1) + C1t - (P2t) - (DP1t \times 3) \quad t = 1, \dots, T \quad (3)$$

(Estoque do produto C1 atual = Estoque do produto C1 da semana anterior + Quant. comprada de C1 na semana atual - Quant. produzida de P2 na semana atual - (Demanda de P1 na semana atual x 3))

$$EC2t = C2t - (P2t \times 2) \quad t = 1, \dots, T \quad (\text{na primeira semana})$$

$$EC2t = EC2(t-1) + C2t - (P2t \times 2) \quad t = 1, \dots, T \quad (4)$$

(Estoque do produto C2 atual = Estoque do produto C2 da semana anterior + Quant. comprada de C2 na semana atual - (Quant. produzida de P2 na semana atual x 2))

$$\sum_{i=1}^n (Hi \times Pit) \leq HT \quad t = 1, \dots, T \quad (5)$$

(O somatório das horas de produção por unidade total na semana deve ser menor ou igual as horas máximas de trabalho por semana)

$$(C1t + C2t) + CF \leq CFt * 1000000 \quad t = 1, \dots, T \quad (6)$$

(CFt é uma variável binária que indica se houve ou não compra )

$$P1t \geq DP1t - EP1t \quad t = 1, \dots, T \quad (7)$$

(Quant. produzida de P1 por semana deve ser igual ou maior que a quant. de demanda de P1 na semana - quant. de P1 em estoque)

$$P2t \geq (DP1t \times 2) - EP2t \quad t = 1, \dots, T \quad (8)$$

(Quant. produzida de P2 por semana deve ser igual ou maior que a quant. de P2 necessária para suprir a demanda de P1 na semana - quant. de P2 em estoque)

$$C1t \geq (P2t + (DP1 \times 2)) - EC1t \quad t = 1, \dots, T \quad (9)$$

(Quant. comprada de C1 por semana deve ser igual ou maior que a quant. de C1 necessária para suprir a demanda de P1 e a produção de P2 na semana - quant. de C1 em estoque)

$$C2t \geq (P2t \times 2) - EC2t \quad t = 1, \dots, T \quad (10)$$

(Quant. comprada de C2 por semana deve ser igual ou maior que a quant. de C2 necessária para suprir a produção de P2 na semana - quant. de C2 em estoque)

#### Restrições de Domínio:

$$CFt = \{0,1\} \quad (11)$$

$$EPit, ECit, DPit, CFt \in \mathbb{Z} \quad (12)$$

Em relação às restrições, teremos 12 ao total, onde as quatro primeiras são as restrições de estoque, a 5 é a restrição de tempo de produção semanal, a 6 é a restrição de compras dos insumos C1 e C2, da 7 a 10 temos as restrições que garantem que a demanda será suprida, e por fim, 11 e 12 são as restrições de domínio das variáveis.

## 4. Implementação

Inicialmente, todos os dados como demanda, preço para estocar, e fazer pedidos foram atribuídos a um arquivo txt, para desta forma fosse possível analisar o comportamento do plano gerado pelo modelo durante os diferentes dados de entrada, verificando a possibilidade de minimizar os custos. Sendo assim, haverá a inclusão do modelo que calcula o custo para saber se é melhor estocar dependendo da demanda atual ou fazer mais pedidos para garantir uma maior produção.

O formato do arquivo txt ficou da seguinte forma:



8  
400  
350  
120000  
450  
180  
50  
100  
200  
100  
440  
250  
220  
350  
180  
250

Figura 01: Instâncias do arquivo txt

Semanas	Semana 01	Semana 02	Semana 03	Semana 04	Semana 05	Semana 06	Semana 07	Semana 08
Demandas de P1	200	100	440	250	220	350	180	250

Figura 02: Explicação das Instâncias do arquivo txt

Estocagem	P1	P2	C1	C2
Valor Semanal	450	180	50	100

Figura 03: Valores de estocagem dos produtos e insumos do arquivo txt

Pedidos	C1	C2
Valor Por pedido	450	180
Custo Fixo		120.000

Figura 04: Valores dos pedidos dos insumos do arquivo txt

O código implementado começa com a abertura e leitura do arquivo, onde optamos pela leitura linha a linha, devido a forma como padronizamos nossas instâncias. Segue na imagem abaixo essa parte implementada:

```

1 #Lendo arquivos
2 arquivo = open("instancia2.txt", 'r')
3 texto = arquivo.readlines()
4 arquivo.close()
5 print(texto)

['8\n', '40\n', '35\n', '12000\n', '45\n', '18\n', '5\n', '10\n', '200\n', '100\n', '440\n', '250\n', '220\n', '350\n', '180\n', '250']

```

**Figura 05:** Abertura e leitura do arquivo txt

Tendo o arquivo salvo na variável *texto*, o próximo passo foi salvar cada dado em uma variável a qual estava associado. Na **figura 06** podemos observar os dados que foram salvos de acordo com as linhas, com o uso da função *split()*. Foram salvas nessa parte as seguintes variáveis:

- *T*: Que é uma lista que indica as semana, de acordo com o número de semanas informados no txt;
- *CC1*: O custo de compra do insumo C1;
- *CC2*: O custo de compra do insumo C2;
- *CF*: O custo fixo por pedido realizado;
- *VP1*: Valor por unidade do produto P1 em estoque;
- *VP2*: Valor por unidade do produto P2 em estoque;
- *VC1*: Valor por unidade do insumo C1 em estoque;
- *VC2*: Valor por unidade do insumo C2 em estoque;

```

1 #salvando Lista de semanas em T
2 t = int(texto[0].split()[0])
3 T = []
4 for i in range(t):
5     T.append(int(i+1))
6
7 print("Semanas:", T)
8
9 #Salvando preços de C1, C2
10 CC1 = int(texto[1].split()[0])
11 CC2 = int(texto[2].split()[0])
12 print("Custo de compra de C1:", CC1, "\nCusto de compra de C2:", CC2)
13
14 #Salvando custo fixo por pedido
15 CF = int(texto[3].split()[0])
16 print("Custo fixo por pedido:", CF)
17
18 #Salvando custos de estoques
19 VP1 = int(texto[4].split()[0])
20 VP2 = int(texto[5].split()[0])
21 VC1 = int(texto[6].split()[0])
22 VC2 = int(texto[7].split()[0])
23 print("VP1 =", VP1, "\nVP2 =", VP2, "\nVC1 =", VC1, "\nVC2 =", VC2)
24

```

**Figura 06:** Salvando os custos de pedidos e estocagem do arquivo txt

Encerrando a parte de leitura do arquivo txt, foram salvas as demandas como um dicionário, para isso, foi criada uma lista chamada  $d$  para salvar as demandas e em seguida foi criada a variável  $demandas$  para associar a lista das semanas ( $t$ ) a lista das demandas ( $d$ ), e foi obtido o dicionário.

```

25 #Salvando as demandas de P1
26 d = []
27
28 for i in range(8, len(texto)):
29     d.append(int(texto[i].split()[0]))
30
31 for i in range(t):
32     demandas = dict(zip(T, d))
33
34 print("Demandas:", demandas)

```

**Figura 07:** Salvando as demandas do pedido 1 do arquivo txt

Segue um resumo de todos os dados lidos e salvos:

```

Semanas: [1, 2, 3, 4, 5, 6, 7, 8]
Custo de compra de C1: 40
Custo de compra de C2: 35
Custo fixo por pedido: 12000
VP1 = 45
VP2 = 18
VC1 = 5
VC2 = 10
Demandas: {1: 200, 2: 100, 3: 440, 4: 250, 5: 220, 6: 350, 7: 180, 8: 250}

```

**Figura 08:** Exibição dos dados salvos do arquivo txt

O Gurobi Optimization é um solucionador de otimização comercial para programação linear, programação quadrática, programação quadraticamente restrita, programação linear inteira mista, programação quadrática de número inteiro misto e programação quadraticamente com número inteiro misto. Para o nosso problema, ele foi utilizado para resolver as equações e também para minimizar a função objetivo.

Sabendo disso, a segunda parte do código implementado é onde adicionamos nossas variáveis e restrições ao *solver*. Na **figura 09**, podemos ver a parte onde foram definidas as variáveis, foi usada a flag *vtype* para definir as variáveis como inteiras ou binárias.

```

1 #!pip install gurobipy
2 from gurobipy import *
3
4 #modelo
5 model = Model('MPS')
6
7 #criando as variaveis
8 EP1t = model.addVars(T, vtype=GRB.INTEGER, name = 'EP1t')
9 EP2t = model.addVars(T, vtype=GRB.INTEGER, name = 'EP2t')
10 EC1t = model.addVars(T, vtype=GRB.INTEGER, name = 'EC1t')
11 EC2t = model.addVars(T, vtype=GRB.INTEGER, name = 'EC2t')
12 P1t = model.addVars(T, vtype=GRB.INTEGER, name = 'P1t')
13 P2t = model.addVars(T, vtype=GRB.INTEGER, name = 'P2t')
14 C1t = model.addVars(T, vtype=GRB.INTEGER, name = 'C1t')
15 C2t = model.addVars(T, vtype=GRB.INTEGER, name = 'C2t')
16 Cft = model.addVars(T, vtype=GRB.BINARY, name = 'Cft')

```

**Figura 09:** Declarando as variáveis do modelo

Em seguida, foi definida a função objetivo através da função *model.setObjective()*, e setada para minimização a partir da função *GRB.MINIMIZE*. A figura abaixo mostra como essa parte foi implementada:

```

1 #definindo a função objetivo
2 som1c1 = quicksum(CC1*C1t[i] for i in T)
3 som1c2 = quicksum(CC2*C2t[i] for i in T)
4 som1 = som1c1 + som1c2
5
6 som2EP1t = quicksum(VP1*EP1t[i] for i in T)
7 som2EP2t = quicksum(VP2*EP2t[i] for i in T)
8 som2 = som2EP1t + som2EP2t
9
10 som3EC1t = quicksum(VC1*EC1t[i] for i in T)
11 som3EC2t = quicksum(VC2*EC2t[i] for i in T)
12 som3 = som3EC1t + som3EC2t
13
14 z = som1 + som2 + som3
15 model.setObjective(z, GRB.MINIMIZE)

```

**Figura 10:** Inserindo a função objetiva do modelo no código

Após as definições das variáveis e da função objetivo, foram adicionadas as restrições do modelo através da função *model.addConstrs()*. Além disso foi utilizada uma função de otimização de restrições do modelo chamada *model.Optimize()* Podemos ver isso na figura 11.

```

1 #definindo as restrições
2 model.addConstrs(EP1t[i]==EP1t[i-1]+P1t[i]+demanda[i] for i in T if i!=1)
3 model.addConstrs(EP2t[i]==EP2t[i-1]+P2t[i]-(demanda[i]*2) for i in T if i!=1)
4 model.addConstrs(EC1t[i]==EC1t[i-1]+C1t[i]-P2t[i]-(demanda[i]*3) for i in T if i!=1)
5 model.addConstrs(EC2t[i]==EC2t[i-1]+C2t[i]-(P2t[i]*2) for i in T if i!=1)
6
7 model.addConstrs((C1t[i]+C2t[i])+CF <= Cft[i]*1000000 for i in T)
8 model.addConstrs(P1t[i]>=demanda[i]-EP1t[i] for i in T)
9 model.addConstrs(P2t[i]>=(demanda[i]*2)-EP2t[i] for i in T)
10 model.addConstrs(C1t[i]>=P2t[i]+(demanda[i]*2)-EC1t[i] for i in T)
11 model.addConstrs(C2t[i]>=(P2t[i]*2)-EC2t[i] for i in T)
12
13 #resolver:
14 model.optimize()

```

**Figura 11:** Definindo as restrições do modelo no código

Por fim, foi gerado o resultado da função objetivo através da função *model.ObjVal*. Como podemos observar na imagem abaixo:

```

1 print("Valor da função objetivo: "+str(round(model.ObjVal, 2)))
2 for v in model.getVars():
3     print(str(v.varName)+" = "+str(round(v.x,2)))

```

Valor da função objetivo: 7444800.0

**Figura 12:** Resultado da minimização

## 5. Considerações Finais

Após todo estudo feito a respeito do problema proposto foi possível aplicar os conceitos vistos em aulas de minimização da função objetivo, partindo de uma análise dos dados criados mediante o plano mestre de produção, criando as instâncias para

através do código verificar se a solução funciona de acordo com a modelagem feita. Simulando o modelo no código, foi possível chegar ao resultado de um custo mínimo baseado nas instâncias de entradas, aplicando as restrições e a função objetiva, dessa forma, o conteúdo apresentado durante as aulas pode ser posto em prática com o auxílio de pacotes de software como o Gurobi Optimization, o que foi essencial para discorrer o comportamento do problema aqui apresentado.

## 6. Referências Bibliográficas

GUROBI. Disponível em: <https://en.wikipedia.org/wiki/Gurobi>. Acessado em: 19 de julho de 2020.

JÚNIOR, José A. V. A. RODRIGUES, L. H. PIZZATO, F. A CONSTRUÇÃO DO PLANO AGREGADO ESTRATÉGICO DE PRODUÇÃO UMA ABORDAGEM CRÍTICA E OPERACIONAL. VIII Congresso Brasileiro de Custos – São Leopoldo, RS, Brasil, 3 a 5 de outubro de 2001. Disponível em: <https://anaiscbc.emnuvens.com.br/anais/article/download/2933/2933#:~:text=O%20pr%20esente%20artigo%20prop%C3%B5e%20uma,Plano%20Agregado%20Estrat%C3%A9gico%20da%20Produ%C3%A7%C3%A3o>. Acessado em: 16 de Julho de 2021.

LEÃO, Thiago. Plano Mestre de Produção: o que é, para que serve e como montar o seu. Nomus. 2021. Disponível em: <https://www.nomus.com.br/blog-industrial/plano-mestre-de-producao/>. Acessado em 16 de julho de 2021.