



# Instituto Tecnológico de Aeronáutica - ITA

## Divisão de Engenharia Eletrônica

Departamento de Eletrônica Aplicada

### Laboratório de CES-12

Pedro Alves de Souza Neto ([alvesouza.pedro97@gmail.com](mailto:alvesouza.pedro97@gmail.com))

## 1: Descrição da implementação

### Solução

#### BB:

O algoritmo escolhido foi o Branch and Bound, este algoritmo utiliza de recursão para se mover em forma similar ao in-order numa árvore. Explicação do que foi feito no código é que ele recebe o valor que sobra da “mochila” e o index do respectivo item, caso o index não possuir item correspondente ele volta para a chamada recursiva anterior e retorna um “false”. Caso o “value” for zero ele retorna “true”.

Na parte intermediária da função, ele chama duas recursões uma com o “value” subtraído pelo peso do item correspondente (indicando que o item foi tomado), o outro com o próprio value que só será chamada se a primeira recursão resultar em falso. No final ele seta “true” no output do item caso este seja tomado.

Ilustração correspondente ao percurso das recursões:

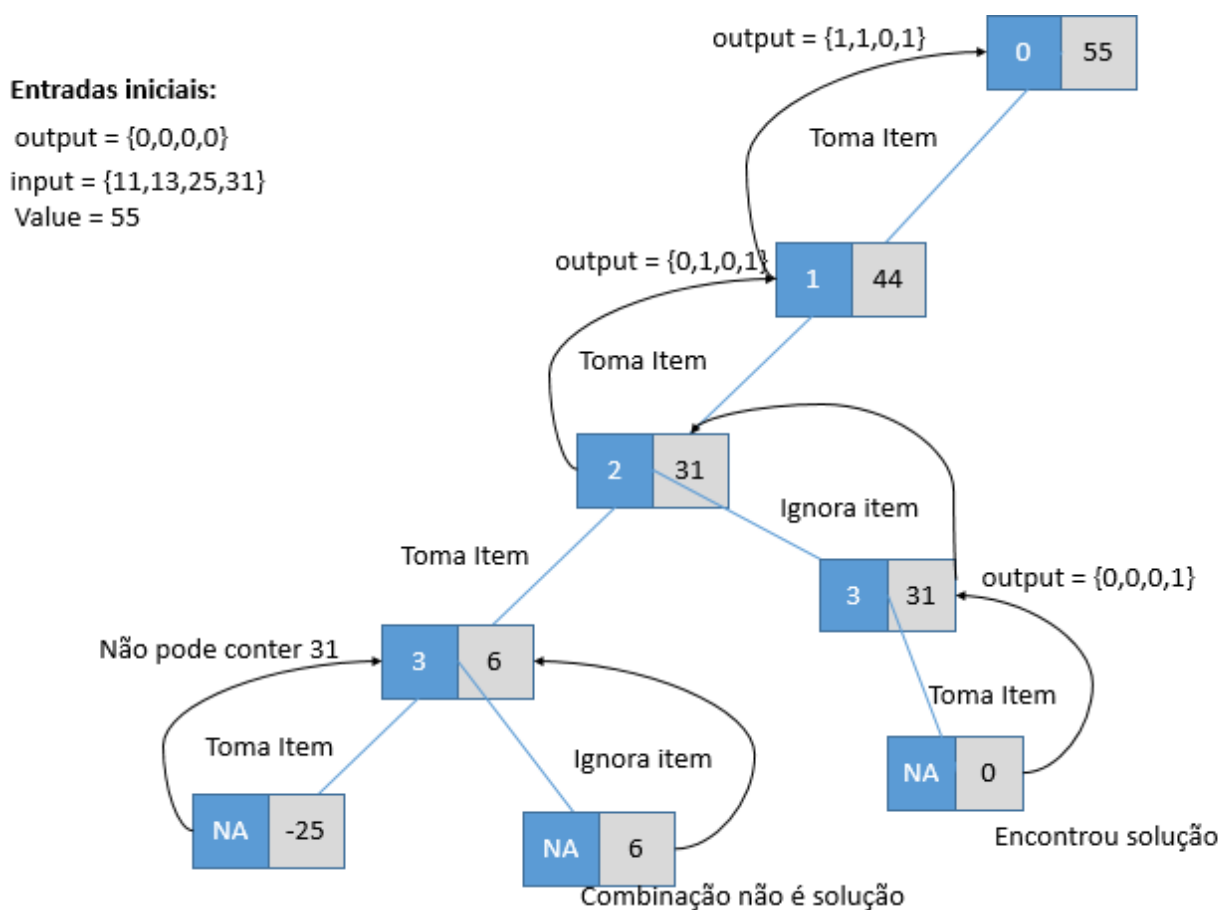


Figura 1.1.Representação do algoritmo.

## PD:

O algoritmo do PD recebe o tamanho da mochila representado por “value”, uma lista de item com seus respectivos pesos, e a lista de output que representa a lista que soluciona o problema.

No começo se cria uma tabela com tamanho  $\text{input.size()} \times (\text{value} + 1)$ , para preencher a tabela com os preços da soma dos itens usados no que no caso deste problema é o peso dos itens.

Para preencher a tabela enquanto o programa está no item “i” ele só poderá usar os valores dos itens  $[0 \ 1 \ 2 \dots i-1 \ i]$ , e o de um ponto da tabela tem que ser menor ou igual ao imediatamente em baixo, caso o peso do item for maior que o tamanho da mochila correspondente ao índice, o valor dessa posição será igual ao imediatamente acima, se não existir o valor será zero.

Caso o peso do item seja menor que peso correspondente ao índice, será escolhido o maior valor entre (soma do preço do item e da combinação que ainda cabe na mochila ou o valor logo encima da posição), este padrão continua até completar a tabela ou encontrar uma solução.

Para se determinar quais os itens escolhidos, parte-se do maior tamanho, no último item, caso a posição acima possui um valor igual ao atual, ele irá subir na tabela, reduzindo os índices. Quando a posição acima for menor que a atual, o item correspondente é acionado no output e reduz o peso da mochila correspondente ao peso do item (fazendo ele se mover pela esquerda).

Entradas iniciais:

output = {0,0,0,0}

input = {1,3,5,7}

Value = 10

Tamanho	0	1	2	3	4	5	6	7	8	9	10
0(1)	0	1	1	1	1	1	1	1	1	1	1
1(3)	0	1	1	3	4	4	4	4	4	4	4
2(5)	0	1	1	3	4	5	6	6	8	9	9
3(7)	0	1	1	3	4	5	6	7	8	9	10

output = {0,0,0,1}

Figura 2.2.Representação do preenchimento da tabela e do output.

## 2: Comparação dos resultados

BB

P3:

Devido a sua recursão tomando um caminho similar ao de uma árvore BB possui um tempo exponencial.

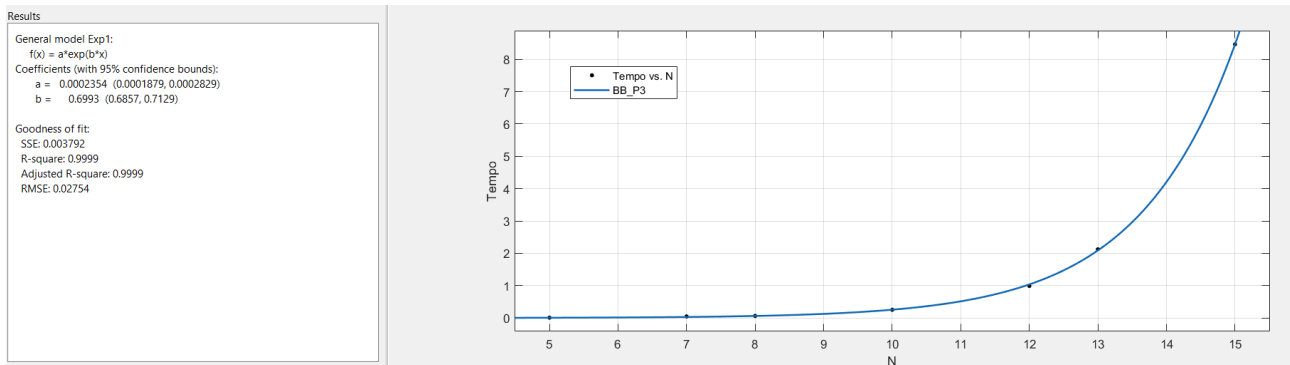


Figura 3.3.Regressão feita pelo MatLab.

## P4:

General model Exp1:  
 $f(x) = a \cdot \exp(b \cdot x)$   
Coefficients (with 95% confidence bounds):  
a = 0.004584 (0.0002129, 0.008956)  
b = 0.4477 (0.3821, 0.5133)  
  
Goodness of fit:  
SSE: 0.06591  
R-square: 0.9941  
Adjusted R-square: 0.993  
RMSE: 0.1148

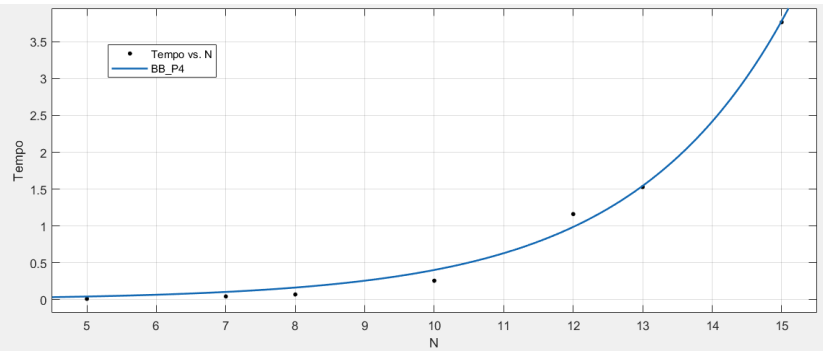


Figura 3.2. Regressão feita pelo MatLab.

## P5:

General model Exp1:  
 $f(x) = a \cdot \exp(b \cdot x)$   
Coefficients (with 95% confidence bounds):  
a = 0.0008393 (0.0002258, 0.001453)  
b = 0.6013 (0.5518, 0.6507)  
  
Goodness of fit:  
SSE: 0.05463  
R-square: 0.9986  
Adjusted R-square: 0.9983  
RMSE: 0.1045

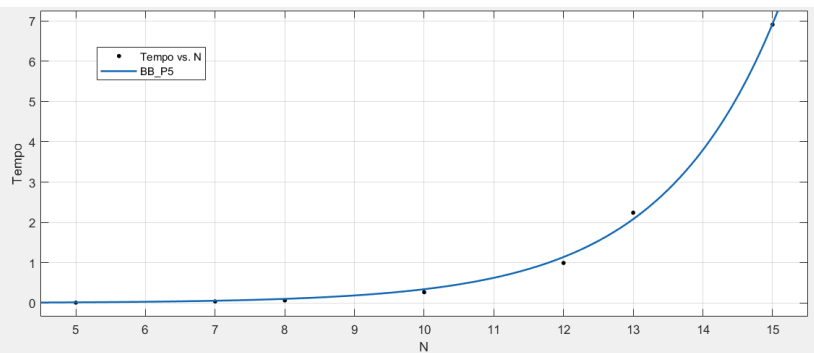


Figura 3.3. Regressão feita pelo MatLab.

## AVIS:

General model Exp1:  
 $f(x) = a \cdot \exp(b \cdot x)$   
Coefficients (with 95% confidence bounds):  
a = 0.0001864 (0.0001111, 0.0002617)  
b = 0.7082 (0.681, 0.7354)  
  
Goodness of fit:  
SSE: 0.01188  
R-square: 0.9997  
Adjusted R-square: 0.9997  
RMSE: 0.04874

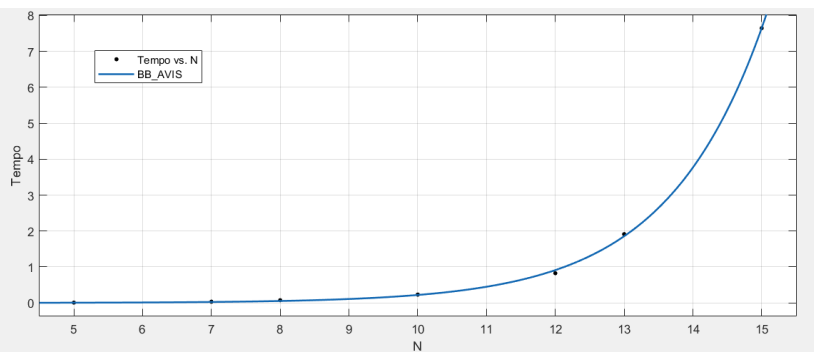


Figura 3.4. Regressão feita pelo MatLab.

## EVOD:

General model Exp1:  
 $f(x) = a \cdot \exp(b \cdot x)$   
Coefficients (with 95% confidence bounds):  
a = 0.0002354 (0.0001879, 0.0002829)  
b = 0.6993 (0.6857, 0.7129)  
Goodness of fit:  
SSE: 0.003792  
R-square: 0.9999  
Adjusted R-square: 0.9999  
RMSE: 0.02754

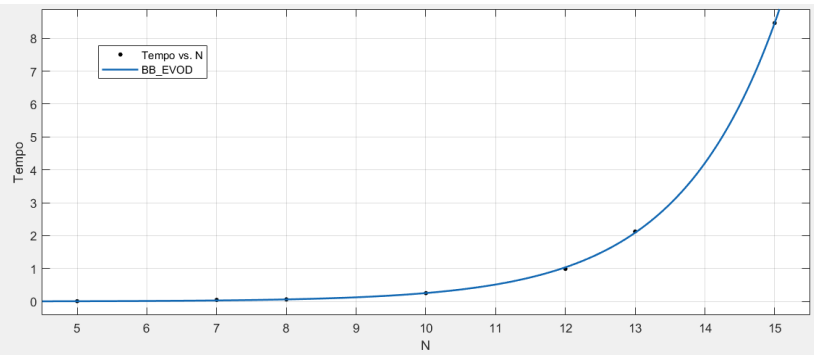


Figura 4.5. Regressão feita pelo MatLab.

## PD

## P3:

Results  
Linear model Poly1:  
 $f(x) = p1 \cdot x + p2$   
Coefficients (with 95% confidence bounds):  
p1 = 0.1594 (0.129, 0.1898)  
p2 = -0.6816 (-1.002, -0.3615)  
Goodness of fit:  
SSE: 0.05317  
R-square: 0.9732  
Adjusted R-square: 0.9679  
RMSE: 0.1031

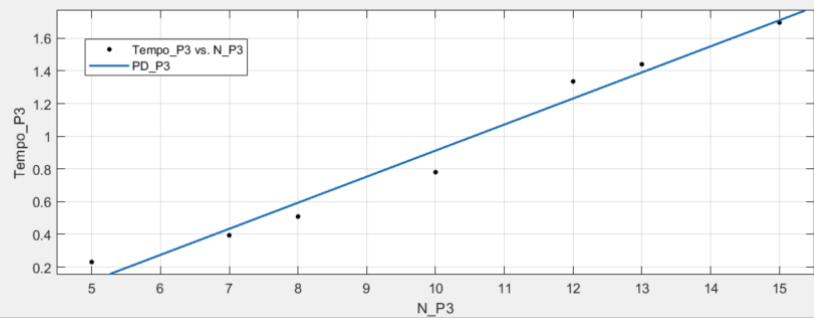


Figura 4.4. Regressão feita pelo MatLab.

## P4:

Results  
Linear model Poly1:  
 $f(x) = p1 \cdot x + p2$   
Coefficients (with 95% confidence bounds):  
p1 = 1.787 (1.541, 2.034)  
p2 = -8.121 (-10.72, -5.526)  
Goodness of fit:  
SSE: 3.493  
R-square: 0.9858  
Adjusted R-square: 0.983  
RMSE: 0.8359

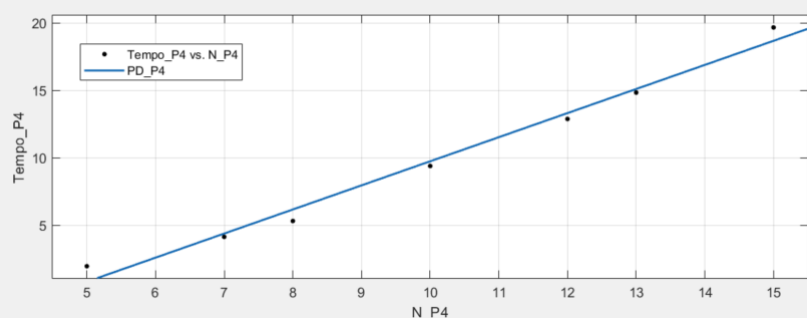


Figura 4.2. Regressão feita pelo MatLab.

P5:

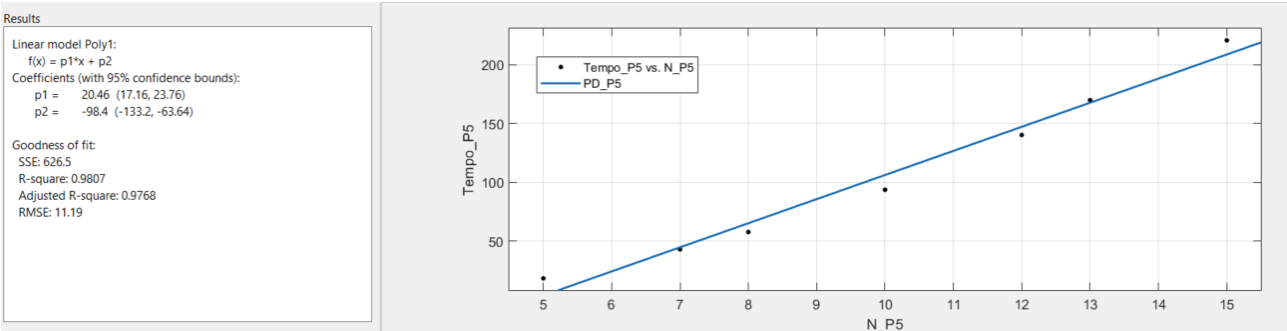


Figura 4.3.Regressão feita pelo MatLab.

AVIS:

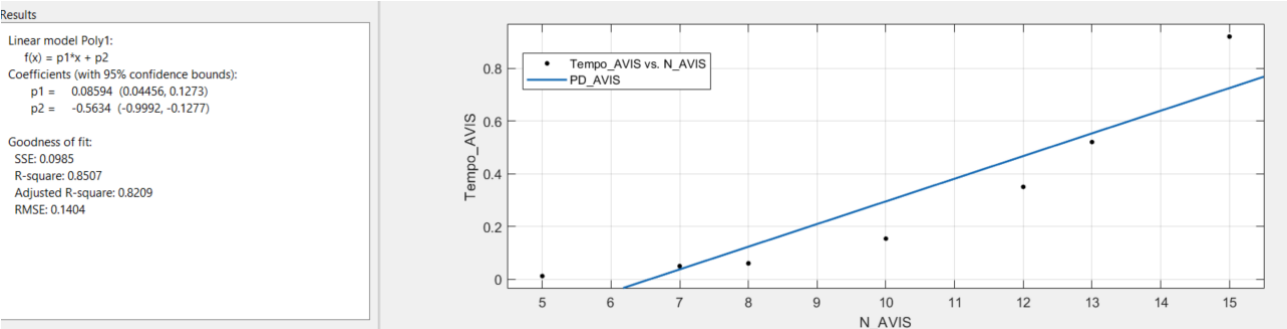


Figura 4.4.Regressão feita pelo MatLab.

EVOD:

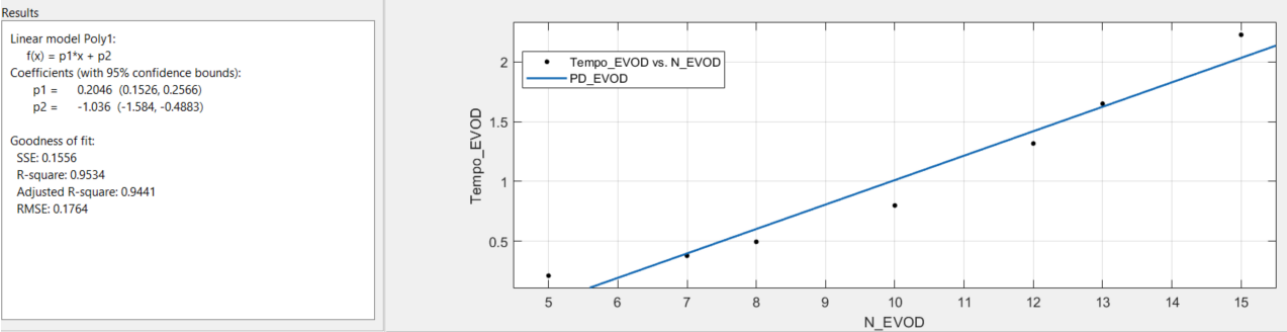


Figura 4.4.Regressão feita pelo MatLab.