

Heuristic methods for the single-machine scheduling problem with periodical resource constraints

Mateus Filipe Moreira Silva - 21.1.4156
Guilherme Salim Monteiro de Castro Paes - 21.1.4109

PROBLEMA 01

IMPLEMENTAÇÃO 02

ÍNDICE

03 ENTRADA 1

04 ENTRADA 2

05 CONCLUSÃO



01

Problema Abordado

Problema Abordado

**Versão do problema
de Programação de
Máquinas**





Problema Abordado



Máquina Única

Uma única máquina para processar as tarefas

Minimização de Makespan

Fazer todas as tarefas o mais rápido possível.

Respeitando as restrições

Dados

Cada Tarefa tem uma quantidade de recurso necessário para ser processada.

Cada Período tem uma quantidade de recurso disponível e um tempo máximo.

02

Implementação



Índices

i

Índices para os períodos
 $\{1, 2, \dots, t\}$

j

Índices para as tarefas
 $\{1, 2, \dots, n\}$

Parâmetros


$$p_j$$

Tempo de
processamento
da tarefa j

$$r_j$$

Quantidade de
Recurso
necessário para
a tarefa j .

$$T$$

Duração máxima
do período

$$R$$

Máximo de
recurso para
cada período

$$M$$

Número Inteiro
suficientemente
Grande



Índices e Parâmetros

```
1  set I;          # INDICE PARA OS PERIODOS
2  set J;          # INDICE PARA AS TAREFAS

8  param p{J};     # TEMPO DE PROCESSAMENTO DE CADA TAREFA
9  param r{J};     # TEMPO DE PROCESSAMENTO DE CADA TAREFA
10 param T;        # DURACAO TOTAL DE CADA PERIODO
11 param R;        # MAXIMO DE RECURSOS DISPONIVEIS POR PERIODO
12 param M;        # NUMERO POSITIVO GRANDE
```

Exemplo de entrada de parâmetros

```
60 data;  
61  
62 set J := 1 2 3 4 5;  
63 set I := 1 2 3;  
64  
65 param p :=  
66     1 2  
67     2 1  
68     3 5  
69     4 4  
70     5 3;
```

```
72 param r :=  
73     1 3  
74     2 3  
75     3 4  
76     4 1  
77     5 1;  
78  
79 param T := 6;  
80 param R := 4;  
81 param M := 1000000;  
82  
83 end;
```

Variáveis de decisão

 x_{ij}

1 Se a tarefa j é processado no período i

0 Caso Contrário

 y_i

1 Se o período i é usado na solução

0 Caso Contrário

 w_i

1 Se esse é o período com maior tempo ocioso

0 Caso Contrário

 z

Computa tempo ocioso no último período

Domínio das Variáveis

$$y_i \geq 0 \quad \forall i$$

Foi Mudada para Binária

$$x_{ij} \in \{0, 1\}, \quad \forall i, j$$

$$w_i \in \{0, 1\} \quad \forall i$$

$$z \geq 0.$$


Variáveis de decisão E Restrições de domínio

```
18 var X{I, J} binary; # 1 SE A TAREFA J E PROCESSADA NO PERIODO I, 0 CASO CONTRARIO
19 var y{I} binary; # 1 SE O PERIODO I E USADO NA SOLUCAO, 0 CASO CONTRARIO
20 var w{I} binary; # 1 SE O PERIODO I E O MAIOR TEMPO OCIOSO , 0 CASO CONTRARIO
21 var z >= 0 ; # COMPUTA O TEMPO OCIOSO (SLACK) DO PERIODO COM MAIOR
```

Função Objetivo

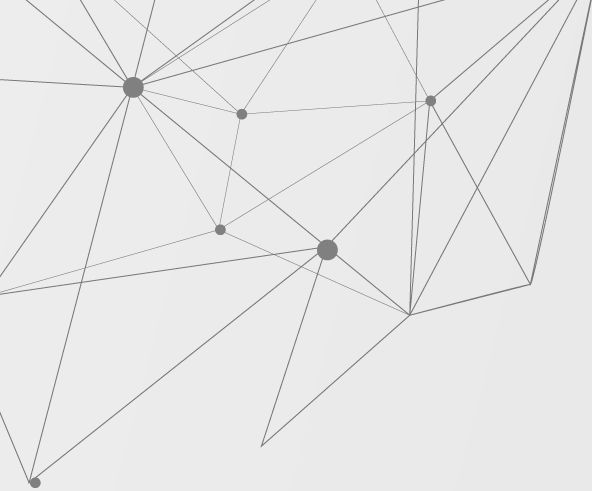
Minimizar
Makespan

$$\underbrace{T}_{\substack{\text{Duração} \\ \text{do período}}} \cdot \underbrace{\sum_{i=1}^t y_i}_{\substack{\text{Quantidade de} \\ \text{períodos usados na} \\ \text{solução}}} - \underbrace{z}_{\substack{\text{Tempo não utilizado} \\ \text{no último período}}}$$



Função Objetivo

```
27 minimize obj:  
28 ... T * sum{i in I} y[i] - z;
```



Restrição 1

$$\sum_{i=1}^t x_{ij} = 1, \quad \forall j$$

Garante que um trabalho seja processado apenas no período i



Restrição 2

$$\sum_{j=1}^n p_j x_{ij} \leq T, \quad \forall i$$

Impõe a restrição de tempo de cada período

Para todo período a soma das tarefas vezes sua respectiva duração deve ser menor ou igual à duração máxima do período.



Restrição 3

$$\sum_{j=1}^n r_j x_{ij} \leq R, \quad \forall i$$

Determina que as tarefas produzidos em um determinado período de produção não exceda o recurso disponível do período.



Restrição 4

$$x_{ij} \leq y_i \quad \forall i, j$$

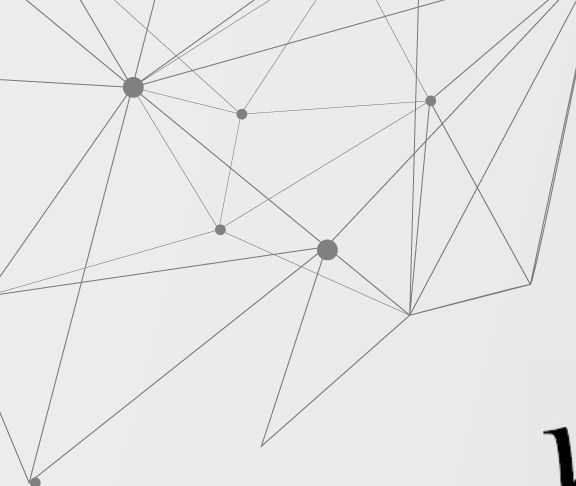
Determina que os trabalhos sejam produzidos apenas nos períodos selecionados.



Restrição 5

$$\sum_{i=1}^t w_i = 1$$

Afirma que apenas um determinado período fornece a folga máxima



Restrição 6

$$w_i \leq y_i \quad \forall i$$

Afirma que, se um período não for utilizado, ele não fornece a folga máxima

Restrição 7

Apenas um período vai
ser diferente de
 $z \leq M$

$$z \leq M(1 - w_i)$$

Período com maior tempo
ocioso
Existe apenas 1
O resto não está na solução
ou não é o Período com
maior tempo ocioso

+

$$Ty_i - \sum_{j=1}^n p_j x_{ij}, \quad \forall i$$

Tempo fixo
Caso o período seja
usado na solução

Calcula o tempo ocioso
nesse período

Calcula o tempo gasto
pelas tarefas desse
determinado período i

Se y_i for 0 esse somatório
também será 0



Restrições

```
34 subject to designacaoTarefas{j in J}:  
35     sum{i in I} X[i, j] = 1;  
36  
37 subject to limiteDeTempo{i in I}:  
38     sum{j in J} p[j] * X[i, j] <= T;  
39  
40 subject to limiteDeRecurso{i in I}:  
41     sum{j in J} r[j] * X[i, j] <= R;  
42  
43 subject to limitaTarefasParaPeriodoUsados{i in I, j in J}:  
44     X[i, j] <= y[i];
```



Restrições

```
46 subject to apenasUmPeriodoComMaiorTempoOcioso:
47     sum{i in I} w[i] = 1;
48
49 subject to limitaPeriodoComMaiorTempoOciosoParaPeriodosUsados{i in I}:
50     w[i] <= y[i];
51
52 subject to calculaMaiorTempoOcioso{i in I}:
53     z <= (M * (1 - w[i])) + (T * (y[i])) - (sum{j in J} (p[j] * X[i, j]));
54
```




03

Entrada 1

Entrada

TAREFA	RECURSO NECESSÁRIO	TEMPO DE PROCESSAMENTO
1	3	2
2	3	1
3	4	5
4	1	4
5	1	3

Duração de cada período

→ 6 Unidades

Número Máximo de recurso por período → 4 Unidades

Saída

TAREFAS

PERÍODO

	1	2	3	4	5
1	0	1	0	0	1
2	0	0	1	0	0
3	1	0	0	1	0

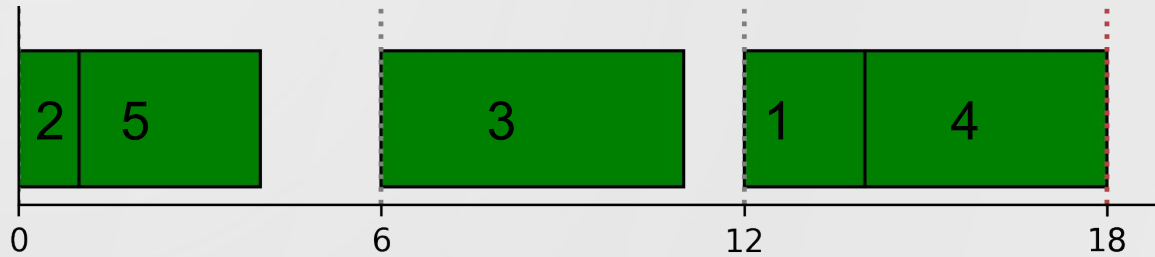
	1	2	3
W	1	0	0
Y	1	1	1

$$Z = 2$$

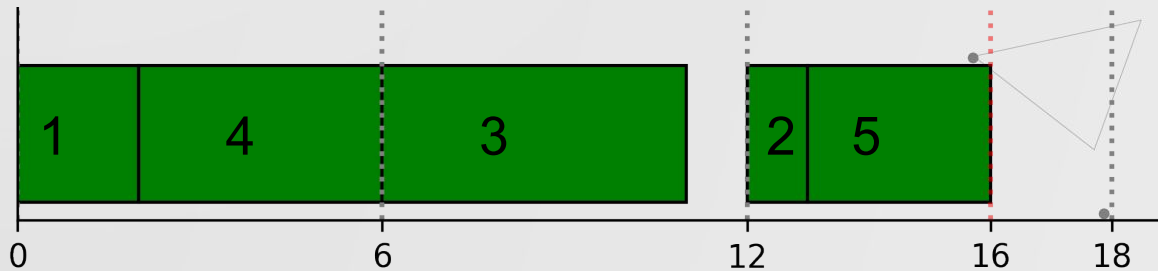
Saída

Função objetivo = 16

Escalonamento das tarefas antes do pós-processamento



Escalonamento das tarefas após ordenas do período com menor tempo ocioso para o maior



04

Entrada 2

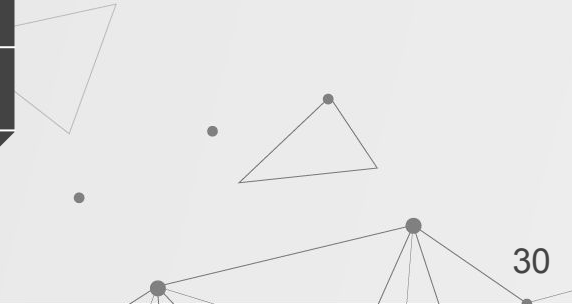


Entrada

TAREFA	RECURSO NECESSÁRIO	TEMPO DE PROCESSAMENTO
1	91	140
2	95	45
3	60	132
4	60	145
5	170	150
6	102	30
7	36	136
8	33	80
9	160	3
10	165	144

Duração de cada
período
188 Unidades

Número Máximo de
recurso por período
170 Unidades



Entrada

PERÍODO

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	1	0	0	0
2	0	0	0	0	0	0	0	0	0	1
3	1	0	0	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	0	1	0
6	0	0	1	0	0	0	0	0	0	0
7	0	0	0	1	0	1	0	0	0	0
8	0	0	0	0	1	0	0	0	0	0

TAREFAS

Saída

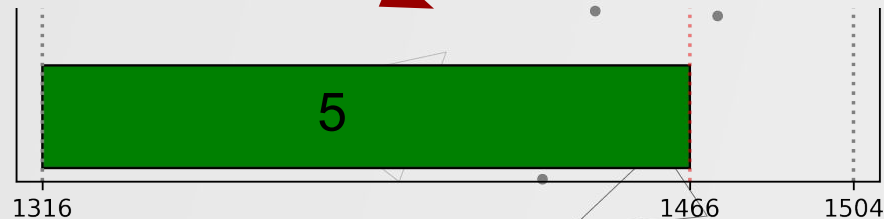
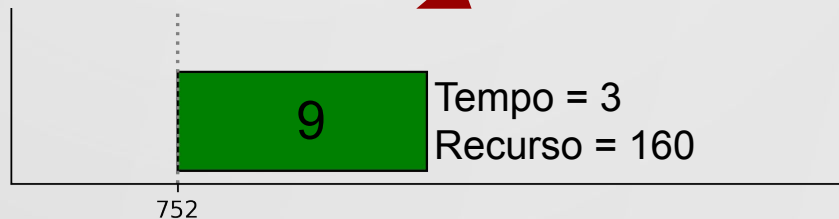
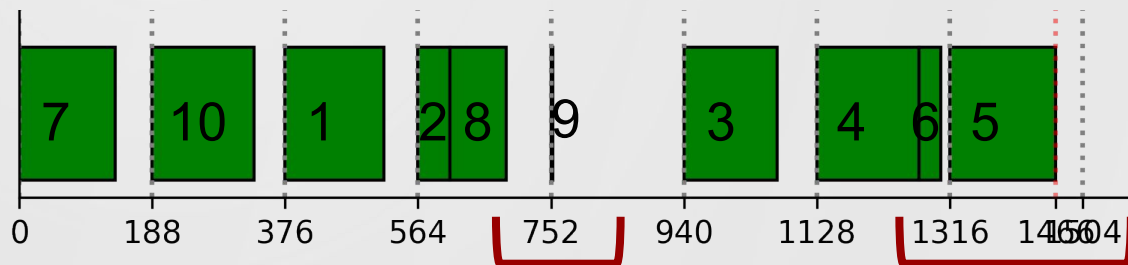
	1	2	3	4	5	6	7	8
w	0	0	0	0	1	0	0	0
Y	1	1	1	1	1	1	1	1

$Z = 185$

Saída

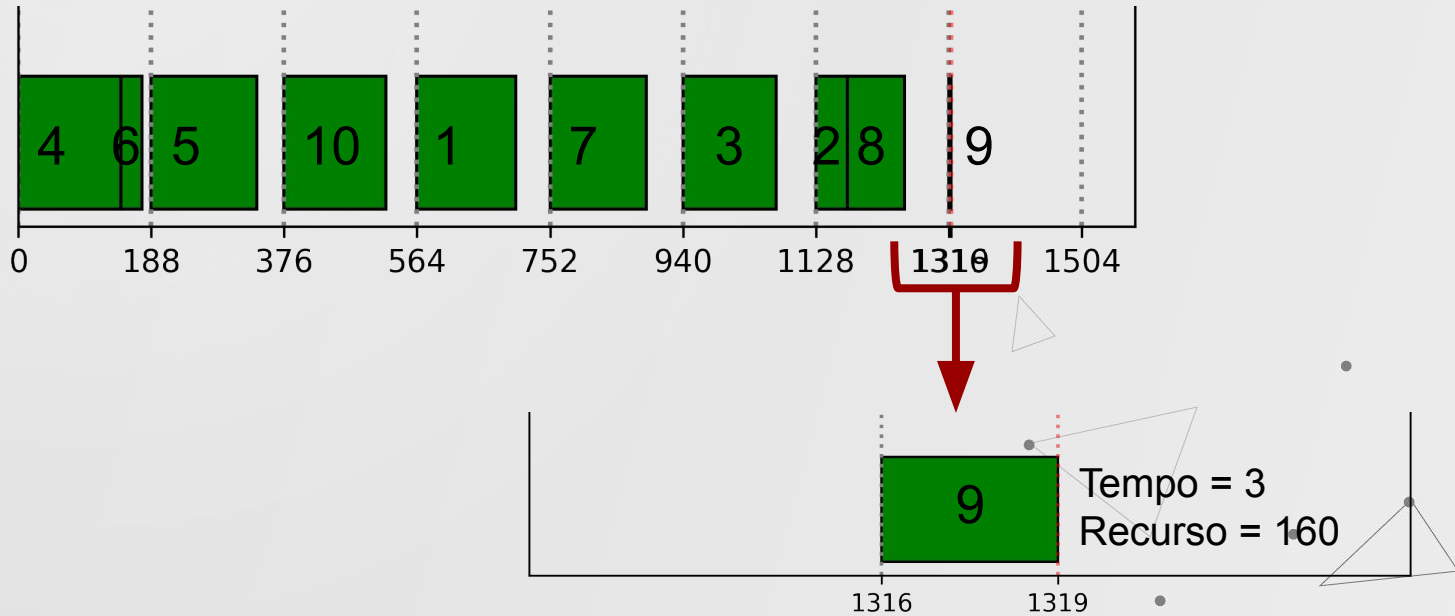
Função objetivo = 1319

Escalonamento das tarefas antes do pós-processamento



Saída

Escalonamento das tarefas após ordenas do período com menor tempo ocioso para o maior



05

Conclusão



Conclusão

- Script em python para fazer pós-otimização dada sequência das tarefas.
- Ordenando as tarefas do período com menor tempo ocioso para o maior.
- Plotar o gráfico usando esses dados e a biblioteca Matplotlib.
- Com restrição de ≥ 0 para o Y_i
 - Não executa em tempo hábil para instâncias maiores que 10 tarefas
 - Foi trocada para Binária
 - Mesmo trocando não executa para instâncias maiores que 15



OBRIGADO

Perguntas?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution.