

TRSP - BUSCA

HEURÍSTICA



HELLO!

Meu nome é Nayara

Disciplina: Tópicos Especiais em Otimização

Professor: Tadeu Zubarán



# PROBLEMA

**Instância:** Um conjunto de  $n$  tarefas, cada tarefa  $i$  tem tempo de processamento  $p_i \in \mathbb{N}$ .

**Solução:** Um agendamento (seleção de tempo de início das tarefas), t.q.

$$|s_i - s_j| \geq \min\{p_i, p_j\}$$

**Objetivo:** Minimizar o tempo de término da última tarefa executada.



## SOLUÇÃO PROPOSTA – GULOSA

Vamos lembrar?

- ✗ Selecciona o maior tempo de processamento da instância;
  - Primeira tarefa a ser processada;
  - Inicia no instante 0.
- ✗ Compara todos os  $s_j$  e pega o menor para ser a próxima tarefa executada.
- ✗ Repete o passo anterior pro novo  $s_i$  até que todas as tarefas tenha seu início definido.



## SOLUÇÃO PROPOSTA

0	1	2	3	4
1	5	8	4	5

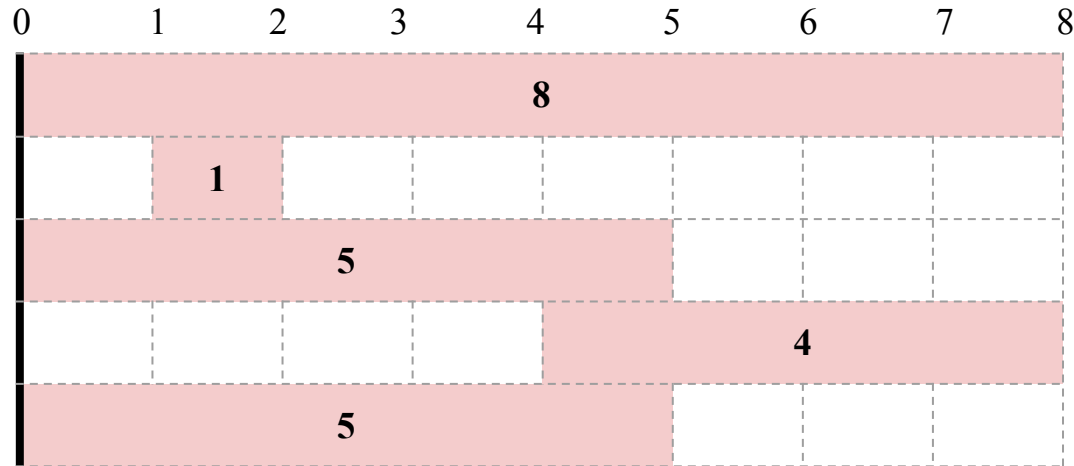
✗ Pegar o maior tempo de processamento para ser o primeiro.

0	1	2	3	4
8	5	1	4	5



## SOLUÇÃO PROPOSTA

✗ Resultado:





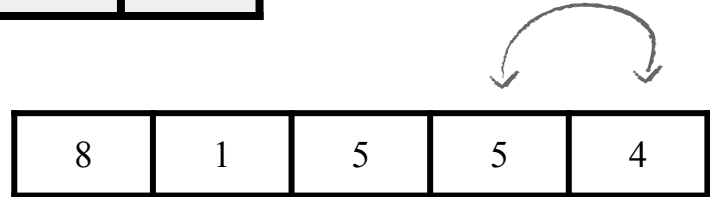
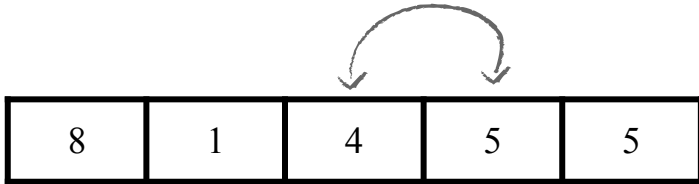
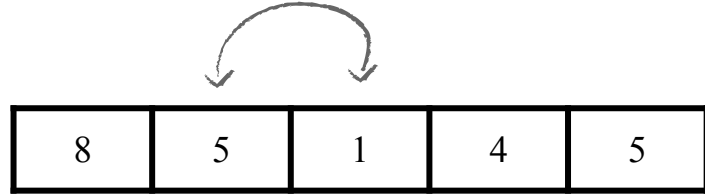
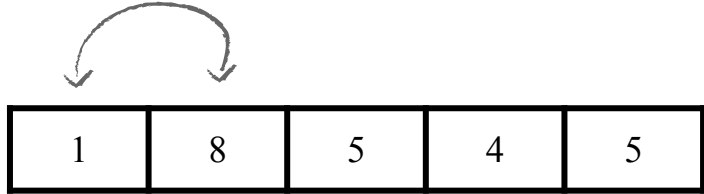
## SOLUÇÃO PROPOSTA – BUSCA LOCAL

- ✗ Pega a solução da heurística construtiva gulosa como solução inicial;
- ✗ Permuta os tempos de processamentos adjacentes, dois a dois;
- ✗ Executa o algoritmo de agendamento em cima da vizinhança atual;
  - Se a solução é melhor, armazena ela;
  - Senão, parte para a próxima vizinhança até achar uma solução melhor.

Não entendeu? Calma, temos um exemplo!



## SOLUÇÃO PROPOSTA – BUSCA LOCAL







## SOLUÇÃO PROPOSTA – BUSCA TABU

- ✗ Pega a solução da heurística construtiva gulosa como solução inicial;
- ✗ Define critério de parada (1 segundo);
- ✗ Define tabu tenure;
- ✗ Cria uma lista tabu  $n \times n$  ( $n$  = tamanho da entrada);
  - Armazena a iteração atual nas posições do swap, se permitido;

Não entendeu? Calma, temos um exemplo!



## SOLUÇÃO PROPOSTA – BUSCA TABU

Iteração  
anterior

8	1	5	4	5
---	---	---	---	---

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	6	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

$(\text{iter} - \text{lista\_tabu}[\text{p1}][\text{p2}]) > \text{tabuTenure?}$

Lista tabu



## LB E DESVIOS PERCENTUAIS

Instância	Limite inferior	Heurística Construtiva	Busca Local	Busca Tabu
50_1	99	Sol: 162 Tempo: 23ms DP: 63,64%	Sol: 162 Tempo: 35ms DP: 63,64%	Sol: 162 Tempo: 1022ms DP: 63,64%
50_2	99	Sol: 261 Tempo: 23ms DP: 163,64%	Sol: 184 Tempo: 29ms DP: 85,86%	Sol: 153 Tempo: 1023ms DP: 54,54%
50_3	97	Sol: 250 Tempo: 26ms DP: 157,73%	Sol: 144 Tempo: 33ms DP: 48,45%	Sol: 134 Tempo: 1029ms DP: 38,14%
50_4	100	Sol: 172 Tempo: 26ms DP: 72%	Sol: 140 Tempo: 36ms DP: 40%	Sol: 140 Tempo: 1021ms DP: 40%



## LB E DESVIOS PERCENTUAIS

Instância	Limite inferior	Heurística Construtiva	Busca Local	Busca Tabu
100_1	99	Sol: 256 Tempo: 33ms DP: 158,58%	Sol: 158 Tempo: 45ms DP: 59,59%	Sol: 158 Tempo: 1024ms DP: 59,59%
100_2	100	Sol: 229 Tempo: 32ms DP: 129%	Sol: 182 Tempo: 33ms DP: 82%	Sol: 182 Tempo: 1024ms DP: 82%
100_3	100	Sol: 222 Tempo: 34ms DP: 122%	Sol: 162 Tempo: 34ms DP: 62%	Sol: 167 Tempo: 1022ms DP: 67%
100_4	99	Sol: 202 Tempo: 33ms DP: 104,04%	Sol: 159 Tempo: 32ms DP: 60,61%	Sol: 156 Tempo: 1020ms DP: 57,57%



## LB E DESVIOS PERCENTUAIS

Instância	Limite inferior	Heurística Construtiva	Busca Local	Busca Tabu
1000_1	100	Sol: 190 Tempo: 72ms DP: 90%	Sol: 190 Tempo: 102ms DP: 90%	Sol: 190 Tempo: 1029ms DP: 90%
1000_2	100	Sol: 260 Tempo: 63ms DP: 160%	Sol: 192 Tempo: 109ms DP: 92%	Sol: 196 Tempo: 1024ms DP: 96%
1000_3	100	Sol: 259 Tempo: 71ms DP: 159%	Sol: 196 Tempo: 105ms DP: 96%	Sol: 194 Tempo: 1022ms DP: 94%
1000_4	100	Sol: 237 Tempo: 72ms DP: 137%	Sol: 190 Tempo: 108ms DP: 90%	Sol: 190 Tempo: 1021ms DP: 90%



SOL1: SOL MÉDIA: 225  
DESVIO PERC. MÉDIO: 126,38%

SOL2: SOL MÉDIA: 171,58  
DESVIO PERC. MÉDIO: 72,51%

SOL3: SOL MÉDIA: 168,5  
DESVIO PERC. MÉDIO: 69,37%



## CONCLUSÃO





# THANKS!

## Dúvidas?



Códigos: [github.com/nayaras/TopicosOtimizacao](https://github.com/nayaras/TopicosOtimizacao)



[nayarathaiza@gmail.com](mailto:nayarathaiza@gmail.com)