

# **Algoritmo Genético para o Problema do Trabalho Balanceado**

INF05010 - 2020/1



## Apresentação do Problema

Dados um conjunto de  $n$  tarefas que precisam ser executadas em sequência, um conjunto de  $m$  operadores e uma matriz  $p$ , onde  $p_{ij}$  representa o tempo necessário para o operador  $j$  executar a tarefa  $i$ , ...




...encontre uma partição das  $n$  tarefas em  $m$  intervalos  $[b_k, e_k]$  e uma permutação dos operadores  $\pi$  de forma que

$$T = \max_{j \in [m]} T_j$$

é minimizado, sendo que

$$T_j = \sum_{t \in [b_t, e_t]} p_{t\pi_j}$$



... encontre uma atribuição de tarefas a operadores de tal modo que operadores executem tarefas sequenciais e o tempo gasto pelo operador que gasta mais tempo trabalhando é minimizado.



# Formulação como Programa Inteiro

Variáveis:

- $x_{ij} \in \{0, 1\}$ ,  $\forall i, j \mid i \in [n] \wedge j \in [m]$ , onde

$$x_{ij} = \begin{cases} 1, & \text{Caso tarefa } i \text{ é executada pelo operador } j \\ 0, & \text{Caso contrario} \end{cases}$$

- $w_{ijk} \in \{0, 1\}$ ,  $\forall i, j, k \mid i, j \in [n] \wedge k \in [m] \wedge i \neq j$ , onde

$$w_{ijk} = \begin{cases} 1, & \text{Caso } x_{ik} \wedge x_{jk} \text{ é verdade} \\ 0, & \text{Caso contrario} \end{cases}$$

- $y \in \mathbb{R}$ , onde

$$y = \max \left( \sum_{i \in [n]} p_{ij} \cdot x_{ij}, \forall j \in [m] \right)$$

**Função Objetivo:**

$$\text{Min. } y$$

**Restrições:**

$$\sum_{j \in [m]} x_{ij} = 1, \quad \forall i \in [n] \quad (1)$$

$$\sum_{i \in [n]} x_{ij} \geq 1, \quad \forall j \in [m] \quad (2)$$

$$w_{ijk} \leq (x_{ik} + x_{jk})/2, \quad \forall i, j, k \mid i, j \in [n] \wedge k \in [m] \wedge i < j \quad (3)$$

$$w_{ijk} \geq x_{ik} + x_{jk} - 1, \quad \forall i, j, k \mid i, j \in [n] \wedge k \in [m] \wedge i < j \quad (4)$$

$$w_{ijk} \leq x_{j-1k}, \quad \forall i, j, k \mid i, j \in [n] \wedge k \in [m] \wedge i < j \quad (5)$$

$$y \geq \sum_{i \in [n]} p_{ij} \cdot x_{ij}, \quad \forall j \in [m] \quad (6)$$



# Algoritmo Genético

- Parâmetros
- Codificação de uma solução
- Seleção da população inicial
- Seleção de indivíduos para crossover
- Crossover
- Mutação
- Seleção da nova população
- Critério de parada



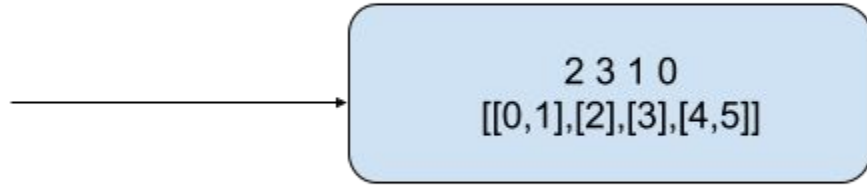
## Parâmetros

$\mu$	Tamanho da população inicial
$\lambda$	Número de indivíduos gerados usando crossover
$\phi$	Probabilidade de indivíduo sofrer mutação
$\omega$	Número máximo de gerações que não melhoram o valor objetivo



## Codificação de uma solução

Operador	Tarefas
0	4, 5
1	3
2	0, 1
3	2





## Seleção da população inicial

- $\mu$  permutações e partições são criadas aleatoriamente e depois são associadas, criando assim  $\mu$  indivíduos



## Seleção de indivíduos para crossover

- São realizados  $\lambda$  k-torneios aleatórios
- O melhor entre os vencedores do torneio é selecionado
- O valor de k foi fixado em 3



## Crossover

$(0\ 1\ 3\ 2\ 4\ 5) \rightarrow (x\ 1\ 3\ 2\ x\ x) \rightarrow (x\ 1\ 3\ 2\ 5\ 4) \rightarrow (0\ 1\ 3\ 2\ 5\ 4)$

$(3\ 1\ 5\ 2\ 4\ 0) \rightarrow (x\ 1\ 5\ 2\ x\ x) \rightarrow (x\ 1\ 5\ 2\ 0\ 3) \rightarrow (4\ 1\ 5\ 2\ 0\ 3)$



## Crossover

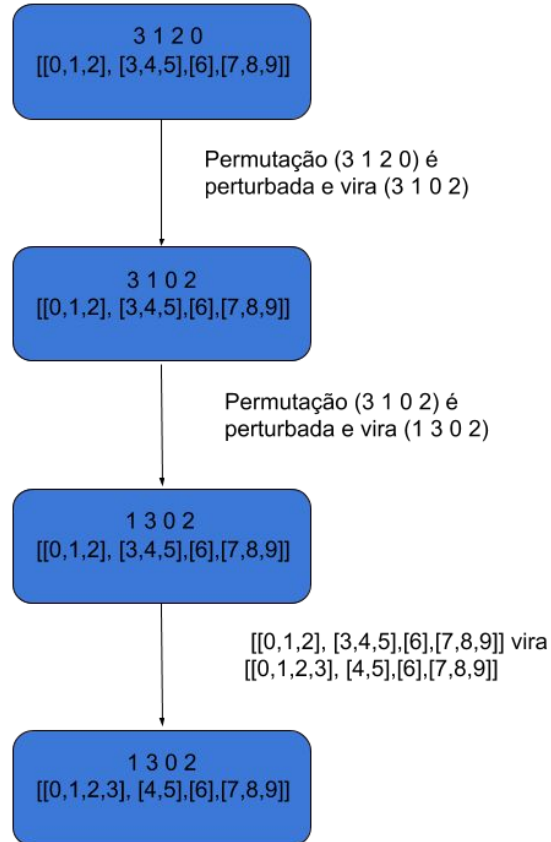
$[[0, 1, 2], [3, 4], [5]]$        $[[0, 1], [2], [3, 4, 5]]$

$$U = \{3, 5\} \cup \{2, 3\} = \{2, 3, 5\}$$

$[[0, 1], [2], [3, 4, 5]]$        $[[0, 1], [2, 3, 4], [5]]$



# Mutação





## Seleção da nova população

- Os  $\lambda$  piores indivíduos entre todos os indivíduos (geração atual e nova geração) são eliminados



## Critério de parada

- $\omega$  gerações consecutivas são criadas e o valor da função objetivo não melhora
- Nenhuma restrição de tempo foi imposta

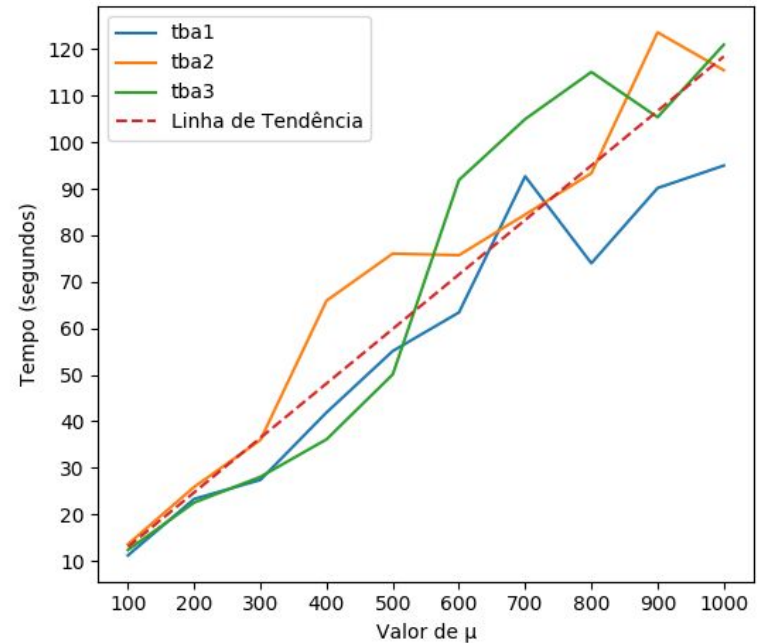
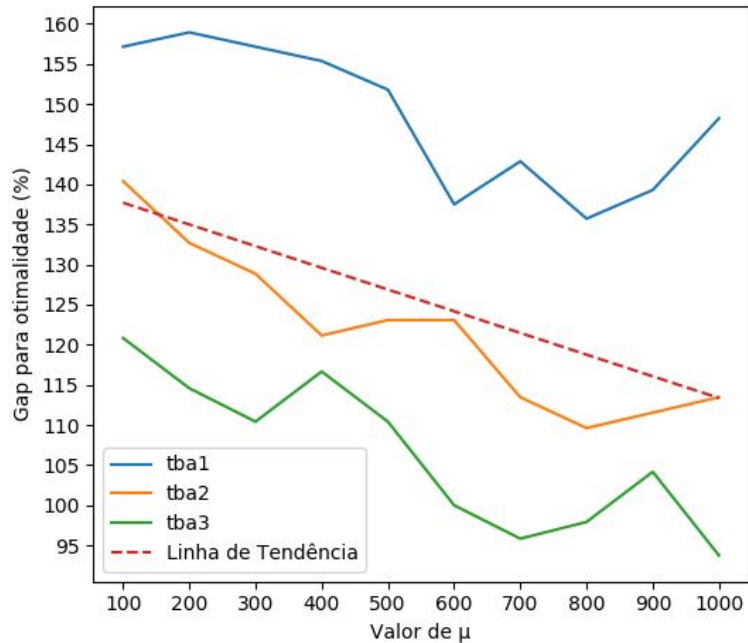




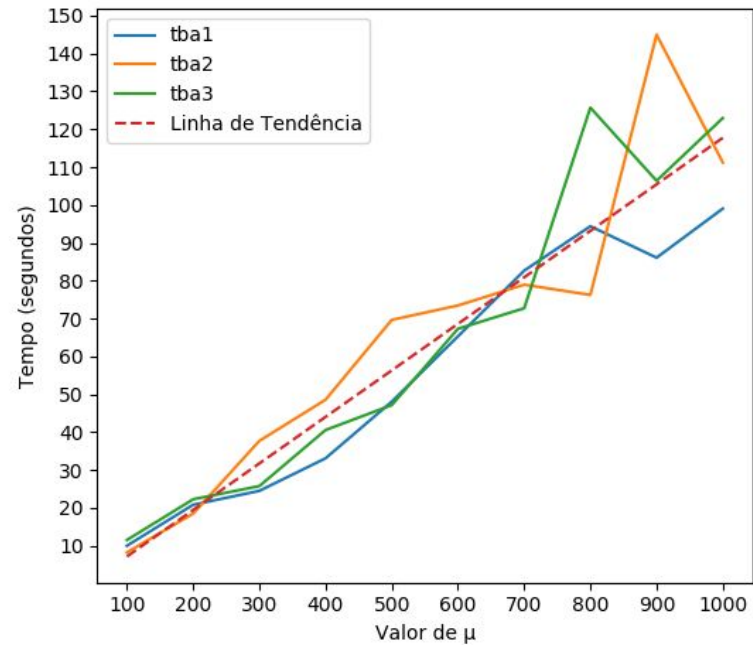
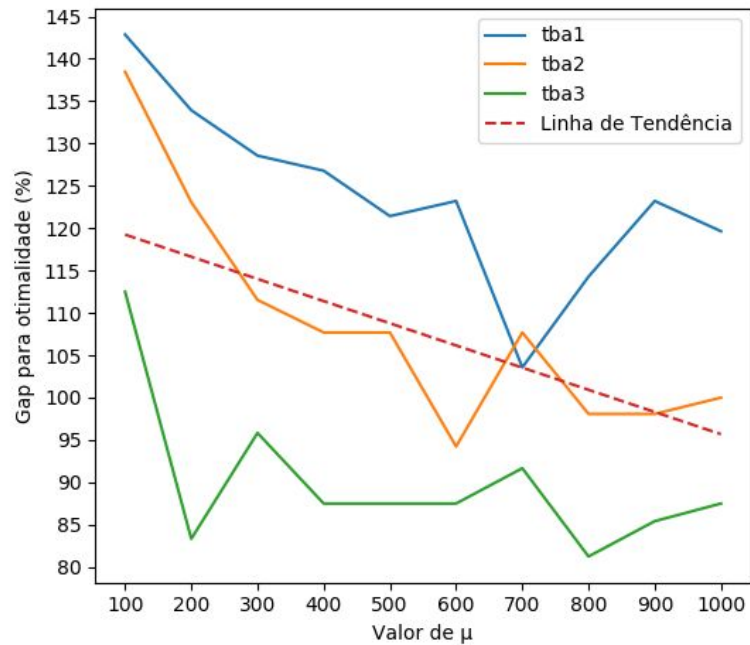
## Resultados Numéricos

- Testes de parâmetros
- Testes de instâncias

# Crossover e mutação desligados



# Crossover ligado

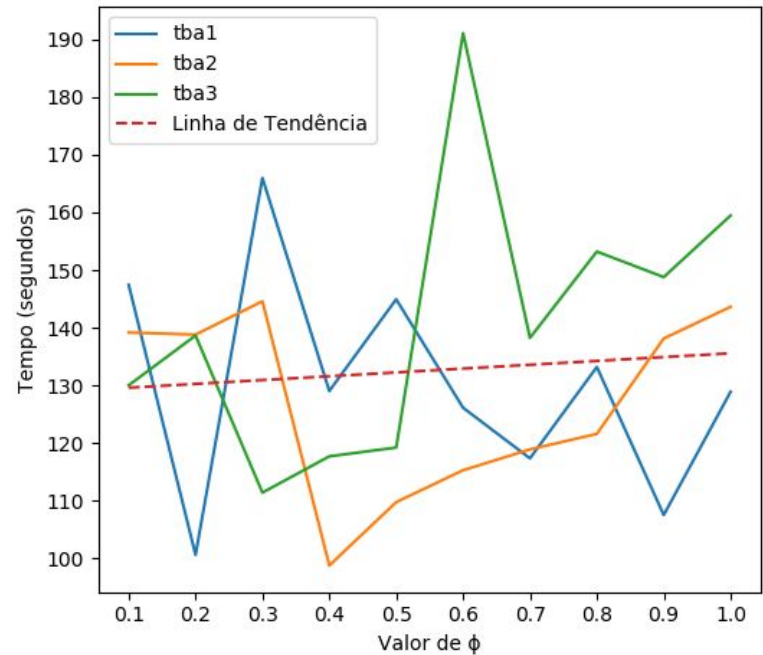
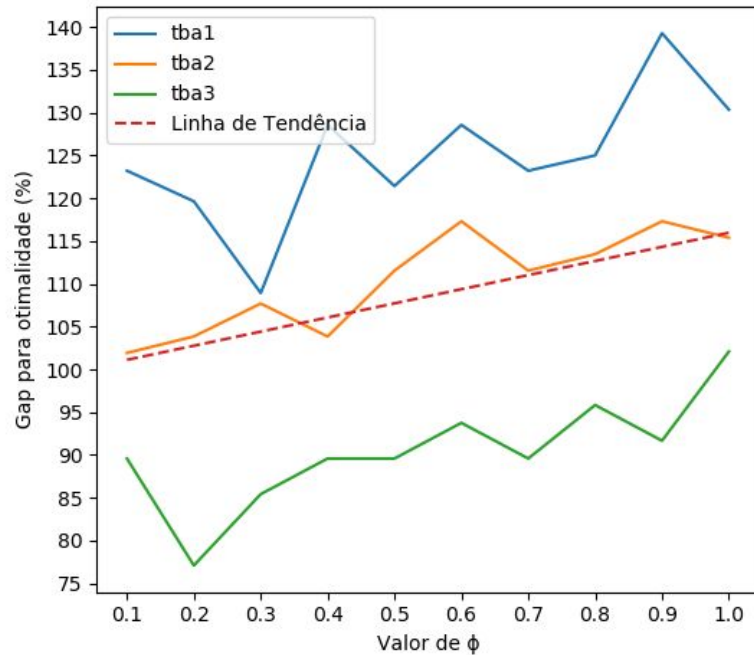




## Impacto do crossover

Instância	Soluções Aleatórias (SA)	Crossover (C)	Desvio de C em relação a SA
tba1	1.39	1.23	11.5%
tba2	1.11	1.04	6.3%
tba3	0.93	0.90	3.2%

# Crossover e mutação ligados

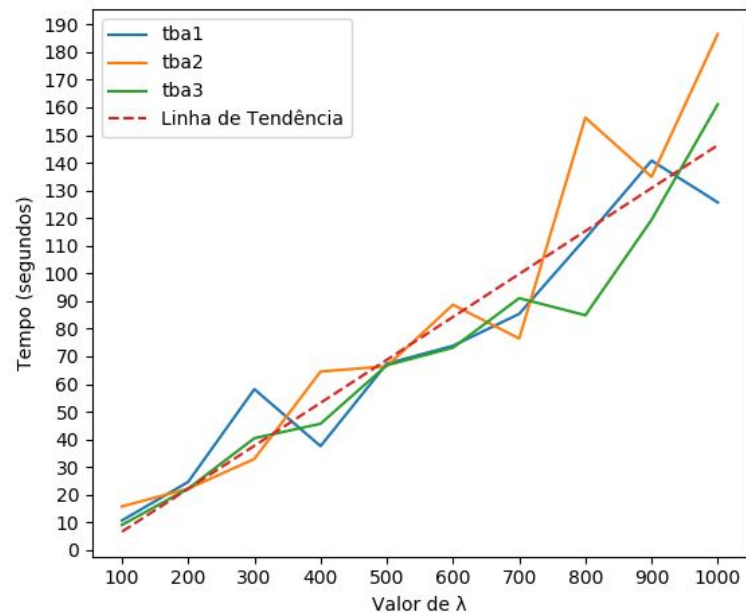
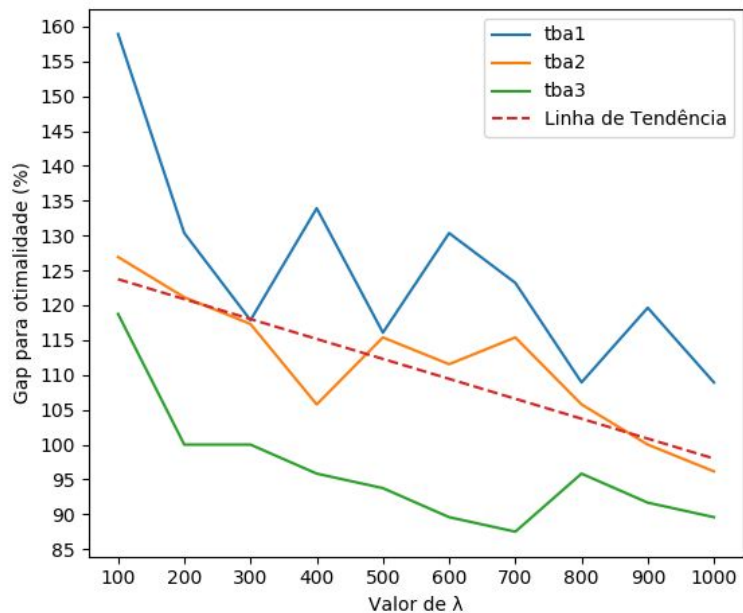




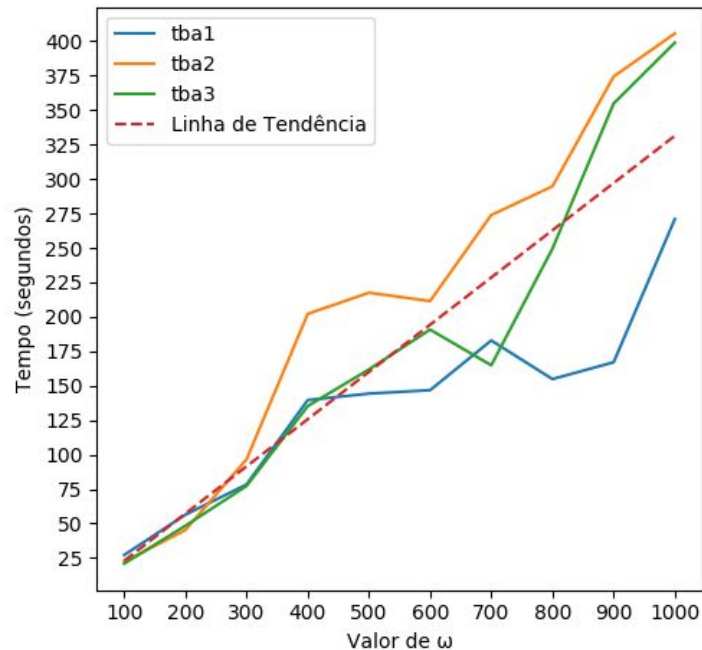
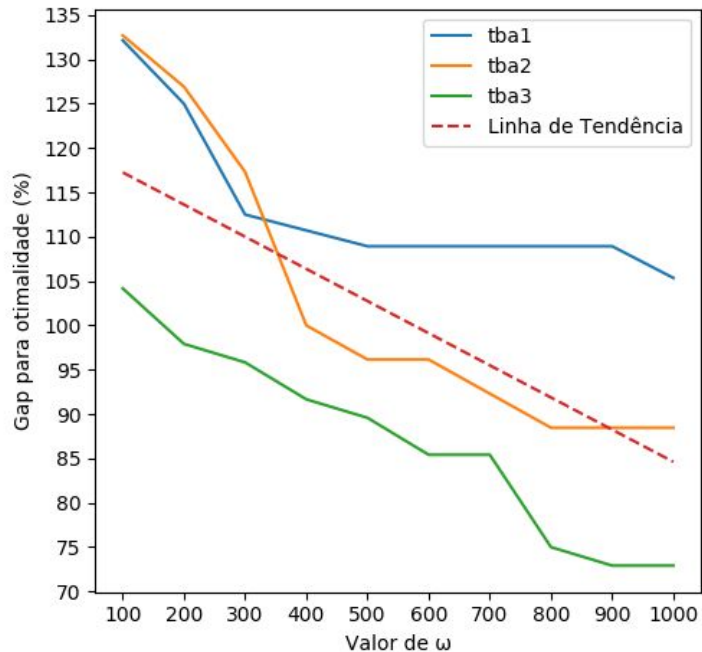
## Impacto de crossover e mutação

Instância	Soluções Aleatórias (SA)	Crossover (C)	Mutação (M)	Desvio de M em relação a SA	Valor de $\phi$
tba1	1.39	1.23	1.17	15.8%	0.3
tba2	1.11	1.04	1.05	5.4%	0.1
tba3	0.93	0.90	0.85	8.6%	0.3

# $\lambda$ - Tamanho da prole



## $\omega$ - Critério de parada







## Algoritmo Genético Calibrado

Instância	Tempo de execução (segundos)	Número de gerações	Valor da Solução Inicial (SI)	Valor da Solução Final (SF)	Desvio da SF em Relação à SI (%)	Desvio da SF em Relação ao BKV (%)
tba1	213.32	1038.80	1.84	1.17	36.41	-108.93
tba2	178.95	1493.00	1.50	1.02	32.00	-96.15
tba3	132.10	1209.40	1.31	0.91	30.53	-89.5
tba4	124.92	1060.80	0.90	0.49	45.55	-58.06
tba5	117.68	1194.80	2.66	1.86	30.07	-24.83
tba6	123.28	1154.60	1.67	0.95	43.11	-66.67
tba7	121.75	1153.60	1.50	0.94	37.33	-59.32
tba8	105.37	1013.40	1.75	1.12	36.00	-31.76
tba9	96.50	1057.80	1.42	0.70	50.70	-20.69
tba10	154.98	1503.20	2.66	1.85	30.45	-39.10



## Resolução via solver

Instância	Tempo de Execução (segundos)	Valor Obtido	Desvio para BKV (%)
tba1	41.53	0.56	0
tba2	11.23	0.52	0
tba3	3.40	0.48	0
tba4	1.47	0.31	0
tba5	108.17	1.49	0
tba6	15.99	0.57	0
tba7	16.18	0.59	0
tba8	4.57	0.85	0
tba9	1.86	0.58	0
tba10	175.04	1.33	0



## Conclusão

- Os operadores de crossover e mutação têm impacto positivo na qualidade da solução
- Para o conjunto de instâncias usado, a resolução via programação inteira é mais proveitosa