Métodos Heurísticos para o Problema de Escalonamento de Tarefas em uma Única Máquina com Data de Entrega Comum

José Eurípedes Ferreira de Jesus Filho
jeferreirajf@gmail.com

Índice

- Problema
- Heurística
- Busca Local
- Algoritmo Memético
- Resultados
- Conclusões

Problema

- Ambiente com n tarefas a serem escalonadas.
- Apenas uma única máquina para processar todas as tarefas.
- Cada tarefa é formada de apenas uma única operação.
- Todas as tarefas estão disponíveis no início do horizonte de planejamento e possuem uma data comum d de entrega.
- Cada tarefa $i = 1, \dots, n$ possui um tempo de de processamento p_i.
- Cada tarefa $i = 1, \dots, n$ possui um custo de adiantamento (estoque) de α_i unidades por unidade de tempo e custo de atraso de entrega de β_i unidades por unidade de tempo.

Problema

- Para cada operação $i = 1, \ldots, n$, objetiva-se determinar o instante de início de processamento s_i de forma a minimizar a soma dos custos de adiantamento ($\sum \alpha_i (d - C_i)$) e atraso das tarefas ($\sum \alpha_i (C_i - d)$).

Heurística

- Seja o conjunto Ω o conjunto de todas as tarefas ainda não escalonadas.
- Em um primeiro passo, para cada tarefa $i \in \Omega$ calcula-se

$$\overline{\alpha_i \beta_i} = \frac{(\alpha_i + \beta_i)}{2},$$

- Que representa a média dos custos por unidade de adiantamento e atraso de cada tarefa.
- Para selecionar a primeira tarefa λ a ser escalonada, primeiro calcula-se
- $\mu_i = \overline{\alpha_i \beta_i}(\mathrm{p}_i - [\max(\mathrm{p}_i - \mathrm{d}, 0)]) - \beta_i\big(\max(\mathrm{p}_i - \mathrm{d}, 0)\big)$

Heurística

- μ_i indica uma estimativa do "ganho" de escalonar a tarefa $i \in \Omega$ para terminar seu processamento no instante d ou o mais próximo disso.
- Assim, escolhe-se $\lambda = \max(u_i),\ \forall\ i \in \Omega$ com instante de início de processamento $\mathrm{s}_\lambda = \max(d - \mathrm{p}_\lambda, 0)$.
- Para os demais passos, para cada tarefa $i \in \Omega$, os possíveis instantes de início de processamento são
- $s_i' = \begin{cases} e - p_i, caso\ e - p_i \geq 0 \\ -\infty, caso\ contrário \end{cases}$
- ou $\mathrm{s}_i'' = t$, quer dizer, o primeiro caso considera adiantar o processamento da tarefa enquanto o segundo caso considera atrasar o processamento da tarefa. e e t são respectivamente o instante de início de processamento da primeira tarefa no escalonamento e o instante de término de processamento da última tarefa no escalonamento (*earliness* e *tardiness*).

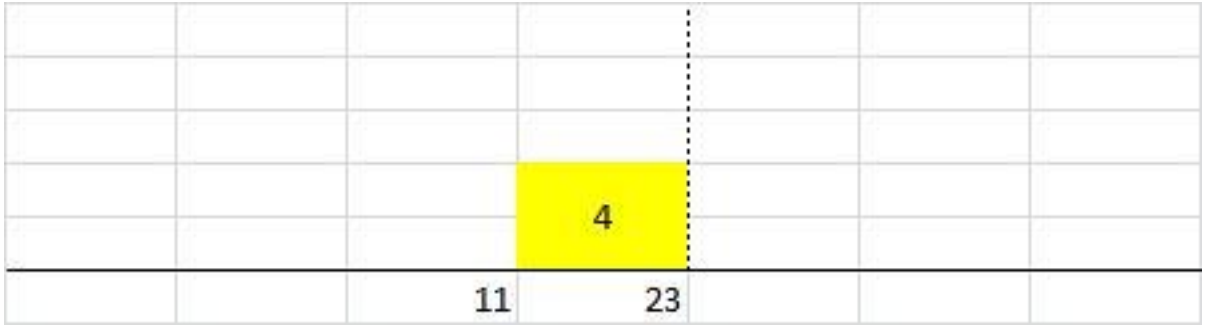
Heurística

- Seja $\omega_1 = \{(i,j) \mid i \in \Omega, j = \min\{\alpha_i(d - s_i'), \beta_i(s_i'' + p_i)\}\}$, quer dizer, ω_1 é o conjunto formado pelos pares (i,j) tais que i é uma tarefa ainda não escalonada e j é o menor custo entre adiantar ou atrasar o processamento de i.
- Seja $j^{max} = \max\{j \mid (*,j) \in \omega_1\}$.
- Seja $\omega_2 = \{i \mid (i,j) \in \omega_1, j = j^{max}\}$.
- Seleciona-se como operação a ser escalonada a operação λ tal que $\lambda \in \omega_2$ e, caso ainda haja mais de um elemento em ω_2, seleciona-se λ tal que $\overline{\alpha_\lambda \beta_\lambda} p_\lambda = k^{max}$, com
- $k^{max} = \max\{\overline{\alpha_i \beta_i} p_i \mid i \in \omega_2\}$.

Heurística

- Exemplo:

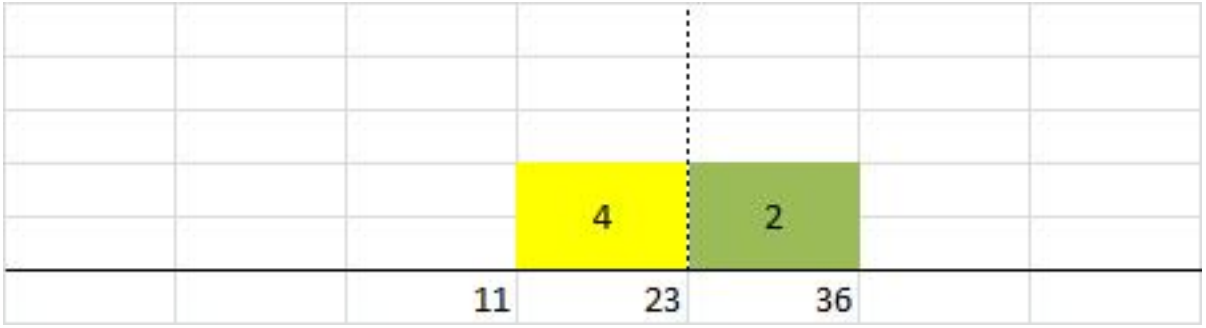
i	1	2	3	4	5
p_i	6	13	13	12	3
α_i	1	5	2	5	6
β_i	15	13	13	15	2
μ_i	48	117	97,5	**120**	12



Heurística

- Exemplo:

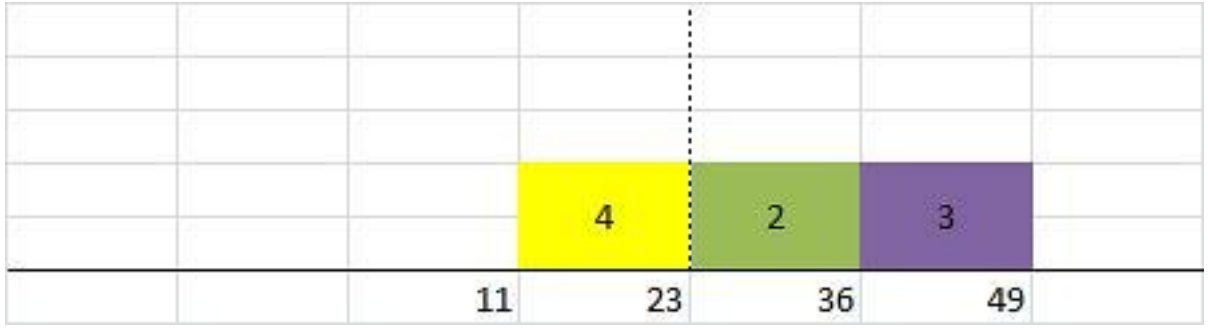
i	1	2	3	4	5
p_i	6	13	13	~~12~~	3
α_i	1	5	2	~~5~~	6
β_i	15	13	13	~~15~~	2
e_i	11	∞	∞	-	72
t_i	90	169	169	-	6
$\min\{t_i, e_i\}$	11	169	169	-	6



Heurística

- Exemplo:

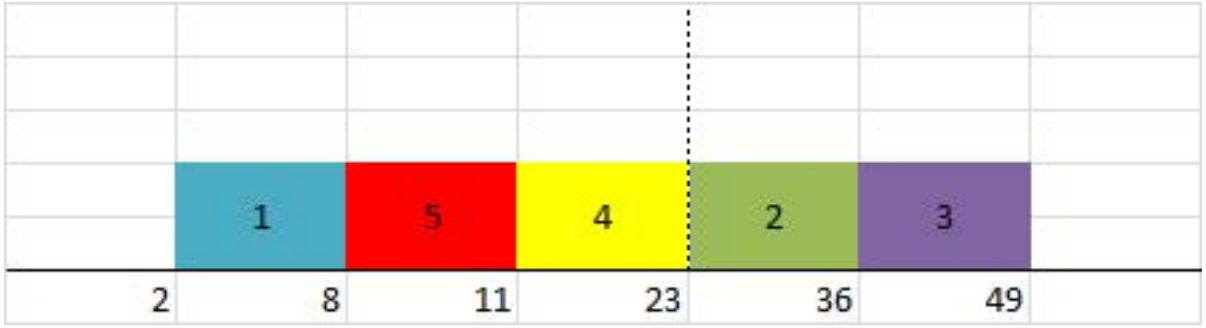
i	1	2	3	4	5
p_i	6	~~13~~	13	~~12~~	3
α_i	1	~~5~~	2	~~5~~	6
β_i	15	~~13~~	13	~~15~~	1
e_i	11	-	∞	-	72
t_i	435	-	468	-	52
$\min\{t_i, e_i\}$	11	-	468	-	52



Heurística

- Exemplo:

i	1	2	3	4	5
p_i	6	13	13	12	3
α_i	1	5	2	5	6
β_i	15	13	13	15	2



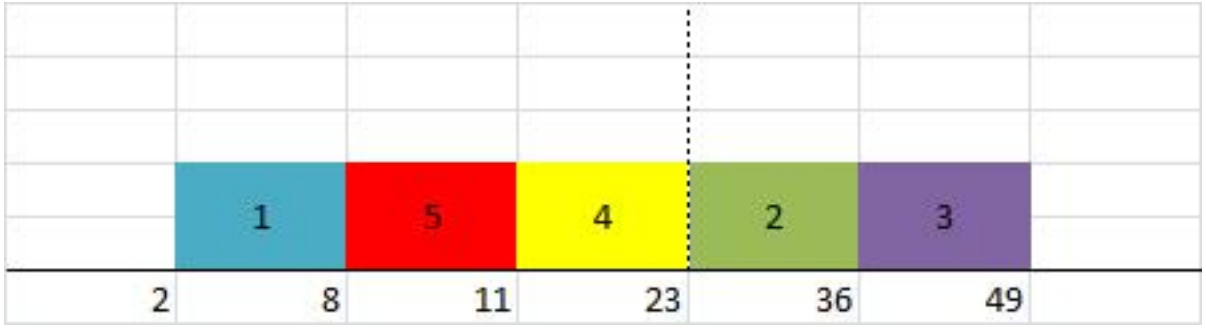
Busca Local

- A heurística construtiva constrói um escalonamento completo e portanto, temos daí o conjunto A e o conjunto B de tarefas adiantadas ou no prazo e atrasadas respectivamente.
- Portanto, a partir do conjunto A constrói-se um vetor binário $\hat{A}$ de tamanho n que diz se a tarefa i, $i = 1, \dots, n$, pertence ($\hat{A}[i] = 1$) ou não pertence ($\hat{A}[i] = 0$) ao conjunto de tarefas adiantadas ou no prazo.

Busca Local

- $n = 5$
- $\hat{A} = \{1, 0, 0, 1, 1\}$
- $\bar{A} = \{1, 5, 4\}$
- $\bar{B} = \{2, 3\}$

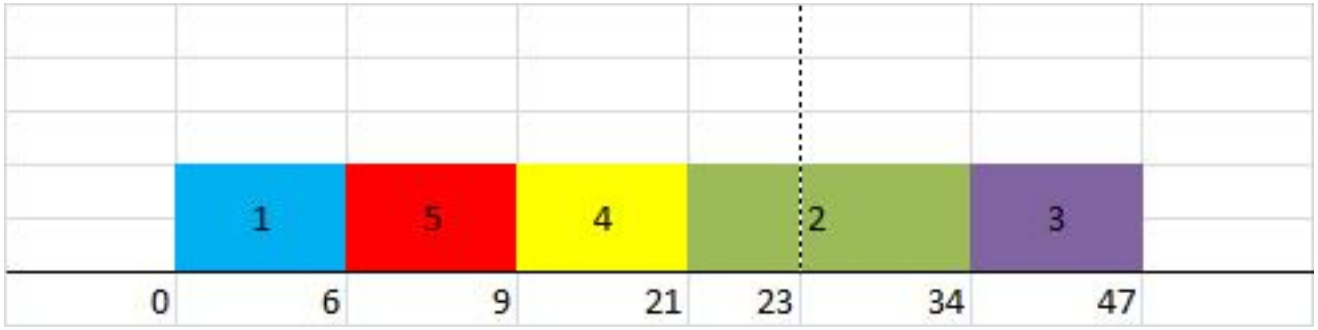
i	1	2	3	4	5
p_i	6	13	13	12	3
α_i	1	6	2	5	1
β_i	15	13	13	15	2



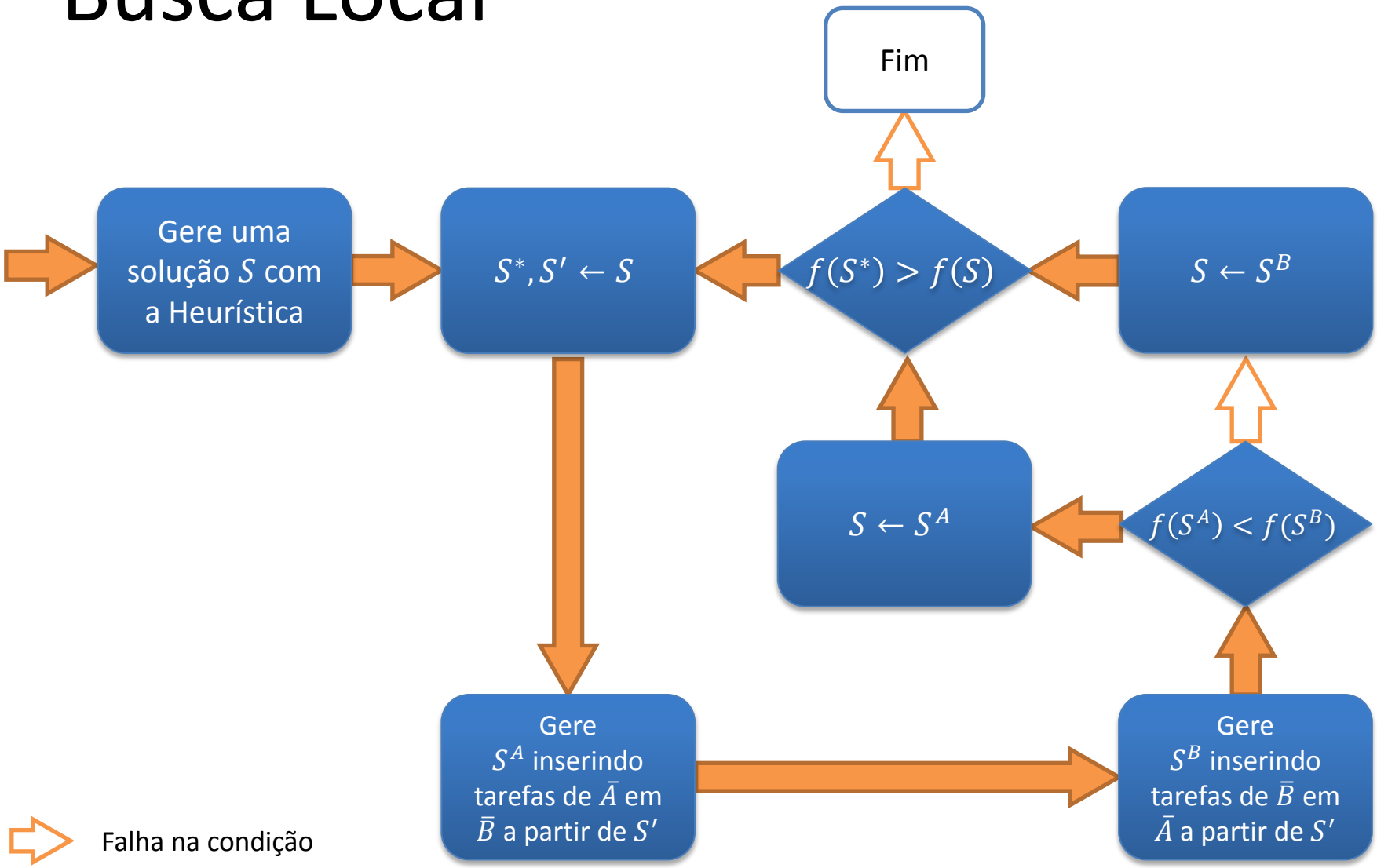
Busca Local

i	1	2	3	4	5
p_i	6	13	13	12	3
α_i	1	6	2	5	1
β_i	15	13	13	15	2
α_i/p_i	0,17	0,46	0,15	0,42	0,33

- Se...
- $n = 5$
- $\hat{A} = \{1, 1,0,1,1\}$
- $\bar{A} = \{1,5,4,2\} \rightarrow \bar{A} = \{1,5,4\}$
- $\bar{B} = \{3\} \rightarrow \bar{B} = \{2,3\}$

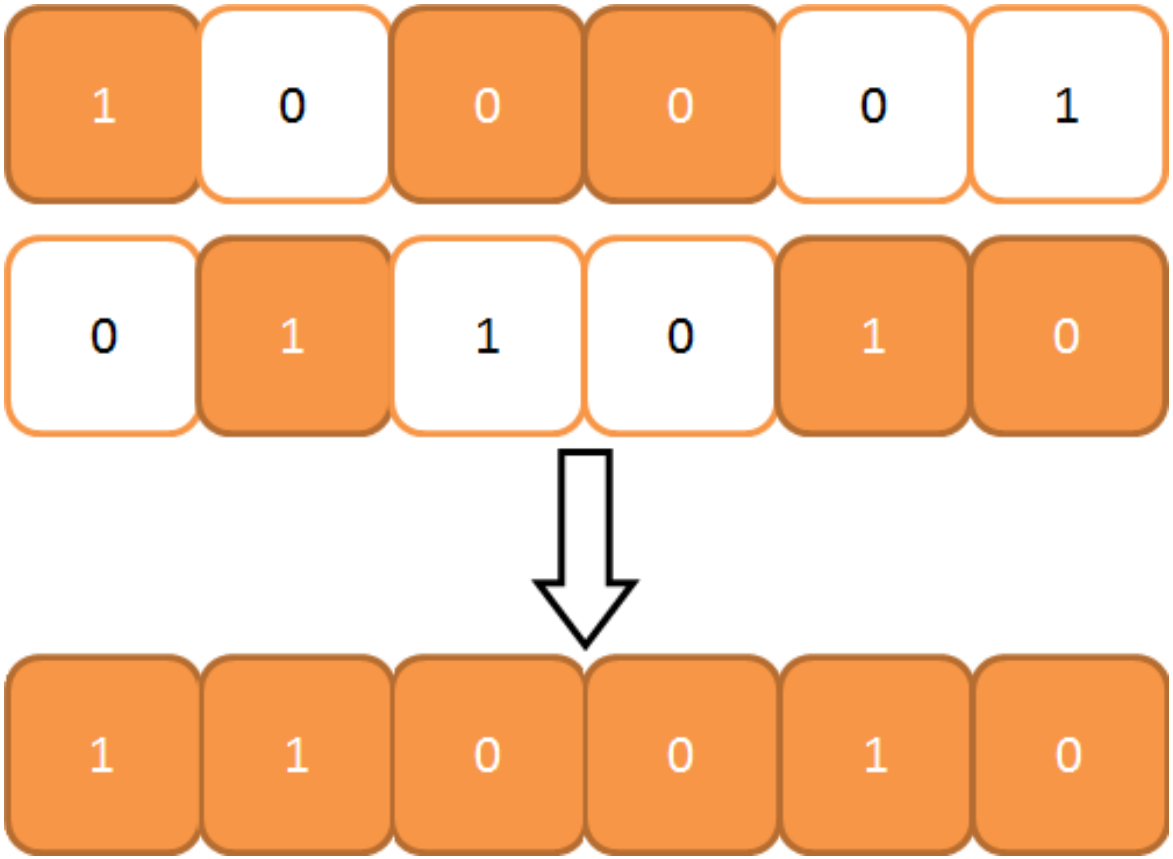


Busca Local


Fim
Gere uma solução S com a Heurística
$S^*, S' \leftarrow S$
$f(S^*) > f(S)$
$S \leftarrow S^B$
$S \leftarrow S^A$
$f(S^A) < f(S^B)$
Gere S^A inserindo tarefas de $\bar{A}$ em $\bar{B}$ a partir de S'
Gere S^B inserindo tarefas de $\bar{B}$ em $\bar{A}$ a partir de S'
Falha na condição

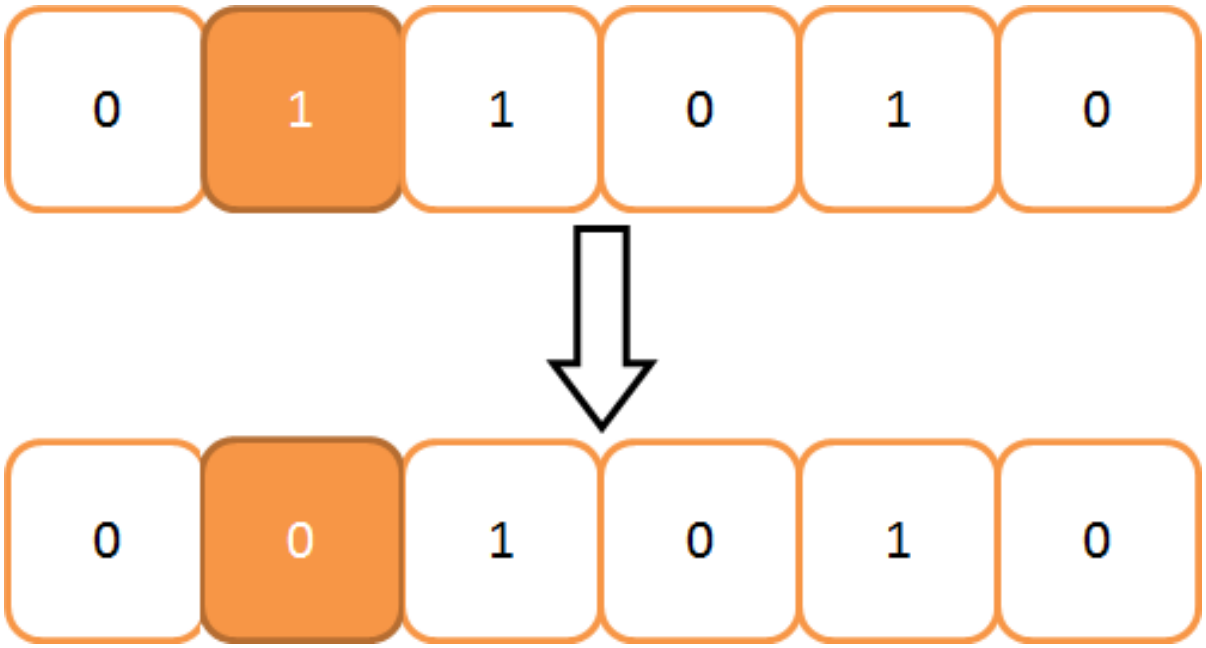
Algoritmo Memético

- Recombinação



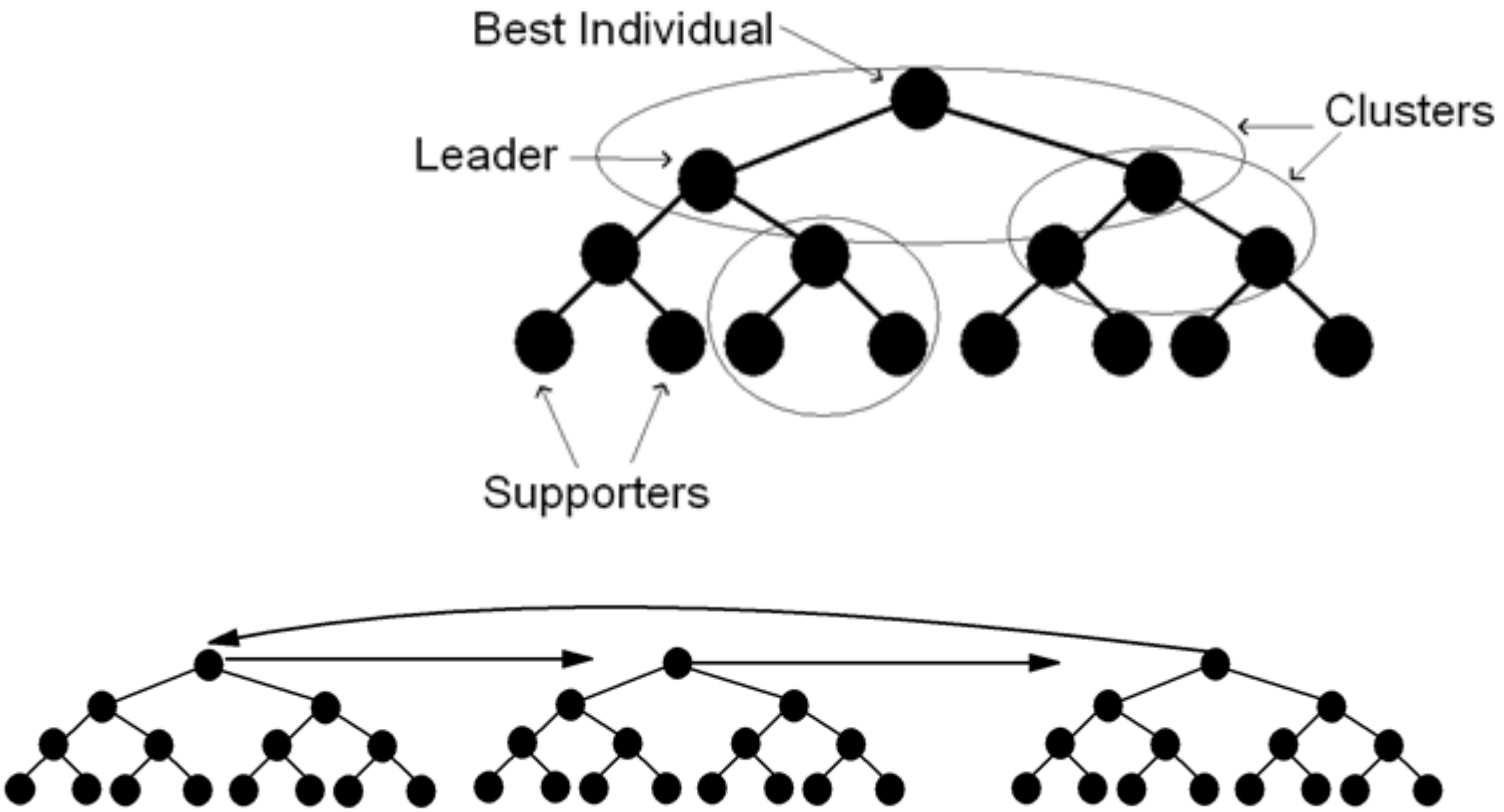
Algoritmo Memético

- Mutação



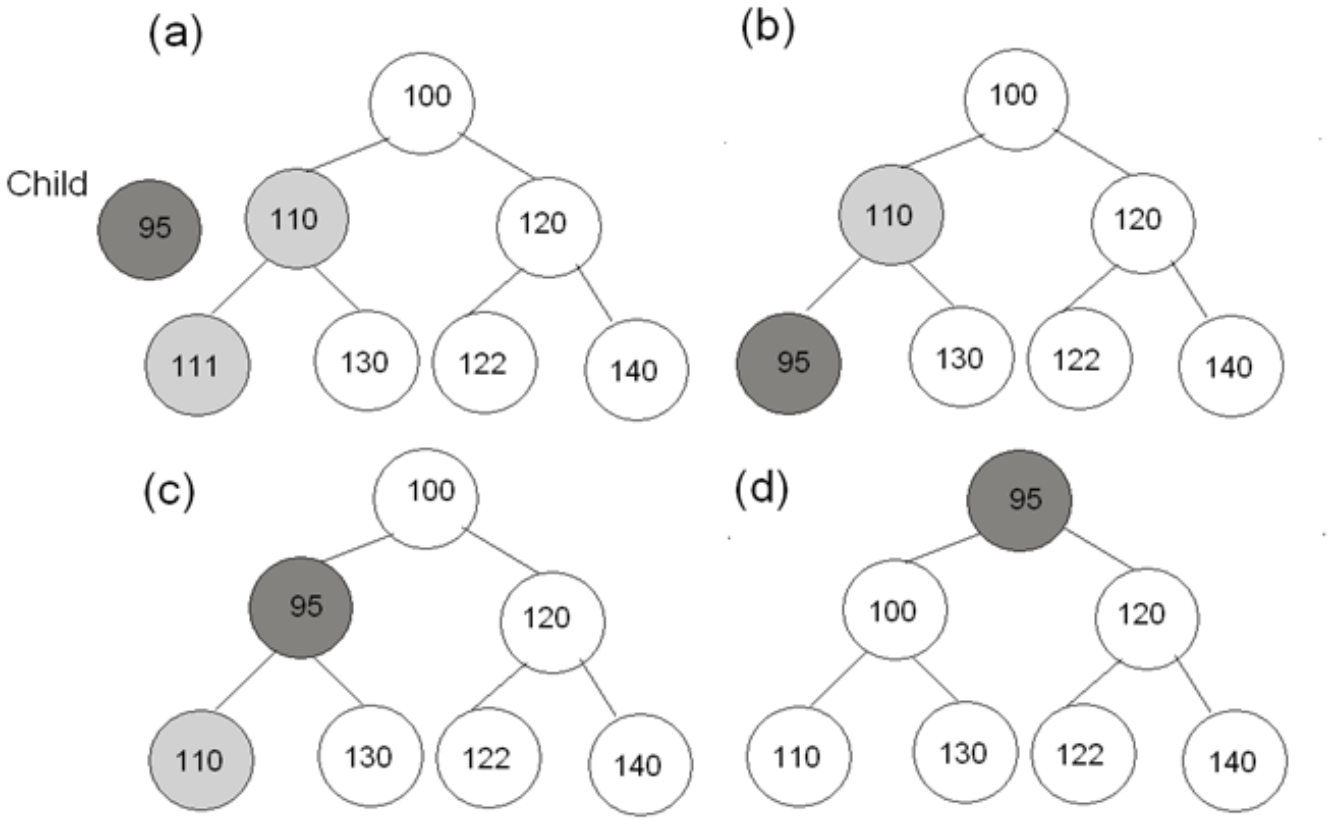
Algoritmo Memético

- Estrutura binária, seleção e migração



Algoritmo Memético

- Reestruturação



Algoritmo Memético

Pseudo-código 1: Descrição do Algoritmo Memético Multi-Populacional Estruturado.

Input: m, γ, λ, ω.
AG$(m, \gamma, \lambda, \omega)$

```
1  begin
2      for i ← 1 to m do
3          initializePop(P_i, ω)
4          evaluatePop(P_i)
5          structurePop(P_i)
6          generation_i ← 0
7      repeat
8          for i ← 1 to m do
9              repeat
10                 for j ← 1 to γ|P_i| do
11                     f_1, f_2 ← selectParents(P_i)
12                     child ← recombine(f_1, f_2)
13                     λ' ∈ [0, 1]
14                     if λ' < λ then child ← mutate(child)
15                     evaluateInd(child)
16                     insertInPop(child, P_i)
17                 structurePop(P_i)
18                 generation_i ← generation_i + 1
19                 if generation_i mod 200 = 0 then localSearchOnTheBestIndividual(P_i)
20             until convergence P_i
21         reinitializePopulations(P_1, ..., m)
22         migrateBestIndividuals(P_1, ..., m)
23     until stop condition
24 end
```

Resultados

- Métodos implementados em C++ e compilados utilizando Microsoft Visual Studio 2008.
- Testes executados em um computador Intel Core i7 4770k, de 3.5Ghz de processamento com 8 GB GDDR3, 1600 Mhz.
- O conjunto de 280 instâncias do *benchmark* proposto por *Biskup and Feldmann (2001)* foi utilizado para avaliar a qualidade dos métodos.

Resultados

- Média dos desvios computacionais obtidos pela heurística construtiva.

	N							
H	**10**	**20**	**50**	**100**	**200**	**500**	**1000**	**Média**
0,2	9,91	10,21	8,14	6,81	6,60	5,07	3,90	7,23
0,4	26,11	28,06	16,03	15,02	15,13	13,84	11,49	17,95
0,6	31,10	15,40	8,31	5,02	3,29	2,38	2,02	9,65
0,8	22,55	11,76	6,61	4,52	3,29	2,38	2,02	7,59
Média	22,42	16,36	9,77	7,84	7,08	5,92	4,86	10,61

Resultados

- Média dos tempos computacionais obtidos pela heurística construtiva (em segundos).

	N							
H	**10**	**20**	**50**	**100**	**200**	**500**	**1000**	**Média**
0,2	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00
0,4	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00
0,6	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00
0,8	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00
Média	0,00	0,00	0,00	0,00	0,00	0,00	0,01	0,00

Resultados

- Média dos desvios computacionais obtidos pela busca local com ordenação de $\bar{A}$ e $\bar{B}$ diferentes varrendo a vizinhança parcialmente.

	N							
H	**10**	**20**	**50**	**100**	**200**	**500**	**1000**	**Média**
0,2	2,93	-1,75	-3,91	-5,39	-3,55	-6,09	-6,75	-3,50
0,4	2,47	1,04	-2,02	-3,19	-2,43	-3,36	-3,90	-1,63
0,6	4,47	0,54	0,48	-0,13	-0,15	-0,11	-0,06	0,72
0,8	2,09	0,63	-0,12	-0,16	-0,15	-0,11	-0,06	0,30
Média	2,99	0,11	-1,40	-2,22	-1,57	-2,42	-2,69	-1,03

- $\bar{A}$ e $\bar{B}$ ordenados de acordo com $\frac{(\beta_i - \alpha_i)}{p_i}$ e $\frac{(\alpha_i - \beta_i)}{p_i}$ respectivamente.
- V Shaped correto.

Resultados

- Média dos tempos computacionais obtidos pela busca local com ordenação de $\bar{A}$ e $\bar{B}$ diferentes varrendo a vizinhança parcialmente (em segundos).

	N							
H	**10**	**20**	**50**	**100**	**200**	**500**	**1000**	**Média**
0,2	0,00	0,00	0,01	0,06	0,37	6,41	55,57	8,92
0,4	0,00	0,00	0,01	0,05	0,32	4,81	32,75	5,42
0,6	0,00	0,00	0,01	0,04	0,22	2,66	17,78	2,96
0,8	0,00	0,00	0,00	0,04	0,22	2,54	17,30	2,87
Média	0,00	0,00	0,01	0,05	0,28	4,11	30,85	5,04

Resultados

- Média dos desvios computacionais obtidos pelas melhores soluções em 5 execuções do AM.

	N							
H	**10**	**20**	**50**	**100**	**200**	**500**	**1000**	**Média**
0,2	0,12	-3,84	-5,69	-6,19	-5,78	-6,43	-6,77	-4,94
0,4	0,19	-1,62	-0,34	-4,94	-3,75	-3,58	-4,40	-2,63
0,6	0,01	-0,72	-4,66	-0,15	-0,15	-0,11	-0,06	-0,83
0,8	0,00	-0,41	-0,24	-0,18	-0,15	-0,11	-0,06	-0,16
Média	0,08	-1,65	-2,73	-2,86	-2,46	-2,56	-2,82	-2,14

Resultados

- Média dos desvios computacionais obtidos pelas médias das soluções em 5 execuções do AM.

	N							
H	**10**	**20**	**50**	**100**	**200**	**500**	**1000**	**Média**
0,2	0,12	-3,84	-5,69	-6,19	-5,78	-6,43	-6,77	-4,94
0,4	0,19	-1,62	-0,34	-4,94	-3,75	-3,58	-4,40	-2,63
0,6	0,01	-0,72	-4,65	-0,15	-0,15	-0,11	-0,06	-0,83
0,8	0,00	-0,41	-0,24	-0,18	-0,15	-0,11	-0,06	-0,16
Média	0,08	-1,65	-2,73	-2,86	-2,46	-2,56	-2,82	-2,14

Resultados

- Média dos desvios computacionais obtidos pelas piores soluções em 5 execuções do AM.

	N							
H	**10**	**20**	**50**	**100**	**200**	**500**	**1000**	**Média**
0,2	0,12	-3,84	-5,69	-6,19	-5,78	-6,43	-6,77	-4,94
0,4	0,19	-1,62	-0,34	-4,94	-3,75	-3,58	-4,39	-2,63
0,6	0,01	-0,72	-4,65	-0,15	-0,15	-0,11	-0,06	-0,83
0,8	0,00	-0,41	-0,24	-0,18	-0,15	-0,11	-0,06	-0,16
Média	0,08	-1,65	-2,73	-2,86	-2,46	-2,56	-2,82	-2,14

Resultados

- Média dos tempos computacionais em segundos obtidos pelo AM considerando a média das 5 execuções.

	N							
H	**10**	**20**	**50**	**100**	**200**	**500**	**1000**	**Média**
0,2	0,00	0,05	1,63	6,36	21,35	171,93	413,88	87,89
0,4	0,00	0,01	2,21	5,56	26,24	150,45	432,45	88,13
0,6	0,00	0,01	0,16	0,69	3,17	30,02	187,07	31,59
0,8	0,00	0,00	0,06	0,87	2,01	36,63	171,33	30,13
Média	0,00	0,02	1,02	3,37	13,19	97,26	301,18	59,43

Conclusões

- Em questão de qualidade de solução o AM se mostrou bastante superior aos demais métodos.
- AM mais caro computacionalmente.
- Quanto a tempo computacional, a heurística construtiva se destacou por gastar, nas instâncias de maior porte, somente 0,01 segundo para construir uma solução.
- A Busca Local fica em posição intermediária, tanto em questão da qualidade da solução quanto do tempo computacional gasto.