

## Soluções 2

Ambiente de execução das soluções: um PC com processador AMD Ryzen 9 39000X com 12 cores de 3.8 GHz, 32 GB RAM, e Ubuntu Linux. Código: [aqui](#).

### Exercício 1 (Busca tabu para o LOP, 5pt)

- a) A busca tabu acabou sendo sensível aos elementos declarados tabu. Assim, escolhi, para um 1-shift da posição  $i$  para a posição  $j$ , declarar ambas as tarefas  $\pi_i$  e  $\pi_j$  tabu e, adicionalmente, se  $i + 2 = j$ , declarar também  $\pi_{i+1}$  tabu. Isso é para evitar desfazer pequenas inversões de tarefas. Além disso, porque a busca do tipo “first improvement” (FI) se mostrou melhor, a busca tabu usa a mesma estratégia; somente caso não vizinhos melhores, o melhor vizinho vai ser selecionado. Ainda o critério de parada é aumentado para  $10n$  iterações sem melhora. O número de replicações é 100 para FI e 10 para busca tabu.
- b) Durações tabu maiores se mostraram melhor, logo testei com

$$d \in D = \{\lceil n/2 \rceil, \lceil n/4 \rceil, \lceil n/8 \rceil, \lceil n/16 \rceil, \lceil n/32 \rceil\}.$$

Para manter o tempo de execução razoável usei instâncias com  $n = 300$ . Tabela 1 mostra os resultados. Podemos ver uma melhora em comparação com a busca FI (na última linha) em todos casos. Para valores melhores de  $d$  os desvios relativos diminuem, e  $d = 0.5$  tem os melhores resultados. A melhora da busca tabu em comparação com FI reflete nos custos: o tempo médio é aproximadamente 17 segundos, duas ordens de grandeza maior que a busca local simples. Por isso o tempo total para obter o melhor desvio relativo sobre todas 100 replicações na busca local FI é comparável com o tempo da busca tabu, e assim melhor que a busca tabu.

- c) A variante com duração tabu aleatória (na linha “Random”) em  $[1, 10]$  foi alterada para durações tabu em  $n[0.75, 1.25]/2$ , mas não consegue melhores resultados.

Tabela 1: Resultados para busca tabu. Numero médio de iterações  $\bar{t}$  e desvio padrão  $\sigma_t$ , número médio de iterações  $\bar{i}$  e desvio padrão  $\sigma_i$ , desvio relativo médio dos melhores valores conhecidos  $\bar{\Delta}$  e desvio padrão  $\sigma_\Delta$ , e melhor desvio relativo dos melhores valores conhecidos "r.d.", em %.

$d$	$\bar{t}$	$\sigma_t$	$\bar{i}$	$\sigma_i$	$\bar{\Delta}$	$\sigma_\Delta$	r.d.
0.0312	17.1	1.1	5,561	256	1.86	0.29	1.42
0.0625	16.0	1.1	5,521	280	1.83	0.28	1.41
0.1250	15.2	2.5	5,820	626	1.80	0.28	1.39
0.2500	21.0	7.6	9,457	2,694	1.75	0.27	1.31
0.5000	15.3	5.3	17,950	5,679	1.48	0.26	1.07
Random	17.3	6.6	18,271	6,286	1.55	0.27	1.14
FI	0.1	0.0	2,611	124	2.09	0.31	1.35

### Exercício 2 (Construção gulosa para o LOP, 5pt)

A primeira observação é que um inserção em uma permutação parcial  $\pi$  pode ser acelerado similar ao aceleração de busca local 1-shift. Ao inserir a tarefa  $t$  na permutação  $\pi = (\pi_1, \dots, \pi_k)$  obtemos permutações  $(\pi_1, \dots, \pi_i, t, \pi_{i+1}, \dots, \pi_k)$  para  $i \in [0, k]$ . Isso aumenta o valor por

$$\Delta(i) = \sum_{l \in [i]} m_{\pi_l, t} + \sum_{l \in [i+1, k]} m_{t, \pi_l}$$

Ainda podemos observar que  $\Delta(i+1) - \Delta(i) = m_{\pi_{i+1}, t} - m_{t, \pi_{i+1}}$  então todos valores  $\Delta(i)$  podem ser encontrados em tempo  $O(k)$ .

Como na questão anterior, os experimentos foram feitos com as 50 instâncias com  $n = 300$ . Testei todos valores  $\alpha = \{0.0, 0.1, \dots, 0.9, 1.0\}$  com 100 replicações (aqui iterações do GRASP). A tabela 2 resume os resultados.

- O algoritmo guloso simples corresponde com  $\alpha = 0$  e coluna  $\bar{\delta}$  na tabela. Podemos ver que a construção é aprox. 5% maior que os melhores valores conhecidos.
- Variando  $\alpha$ , podemos ver na coluna  $\bar{\delta}$  que o valores melhoram para 4.92% com  $\alpha = 0.1$  e depois pioram. O valor  $\alpha = 0.3$  tem um desvio relativo comparável com  $\alpha = 0.0$ , valores maiores, i.e. uma maior aleatoriedade, produz resultados piores. Figura 1 mostra os histogramas. Eles mostram claramente um desvio relativo crescente com  $\alpha$  mas também um aumento da variância, que é melhor para um GRASP.
- O resultado do GRASP corresponde com a coluna  $\bar{\Delta}$  na tabela. Podemos ver que a busca local consegue melhorar por aprox. 4%, em aprox. 6% (o tempo da construção é negligível). Todos resultados são melhores que a busca FI, e busca

Tabela 2: Resultados para GRASP. Tempo  $\bar{t}$ , número iterações  $\bar{i}$ , desvio relativo dos melhores valores conhecidos  $\bar{\Delta}$ , e desvio relativo dos melhores valores conhecidos para construção  $\bar{\delta}$ , em %.

$\alpha$	$\bar{t}$	$\bar{i}$	$\bar{\Delta}$	$\bar{\delta}$
0.0	6.63	100	1.66	5.11
0.1	6.71	100	1.60	4.92
0.2	6.77	100	1.51	4.99
0.3	6.77	100	1.55	5.13
0.4	6.82	100	1.48	5.15
0.5	6.86	100	1.50	5.33
0.6	6.88	100	1.49	5.43
0.7	6.90	100	1.49	5.56
0.8	6.95	100	1.50	5.69
0.9	6.97	100	1.45	5.78
1.0	6.97	100	1.49	5.96

tabus com  $d \leq 0.25$ . A melhor variante do GRASP com  $\alpha = 0.9$  produz um valor médio de 1.45%, o melhor resultado. Porém o melhor valor de 100 busca FI ainda é melhor, mas em aprox. o dobro do tempo.

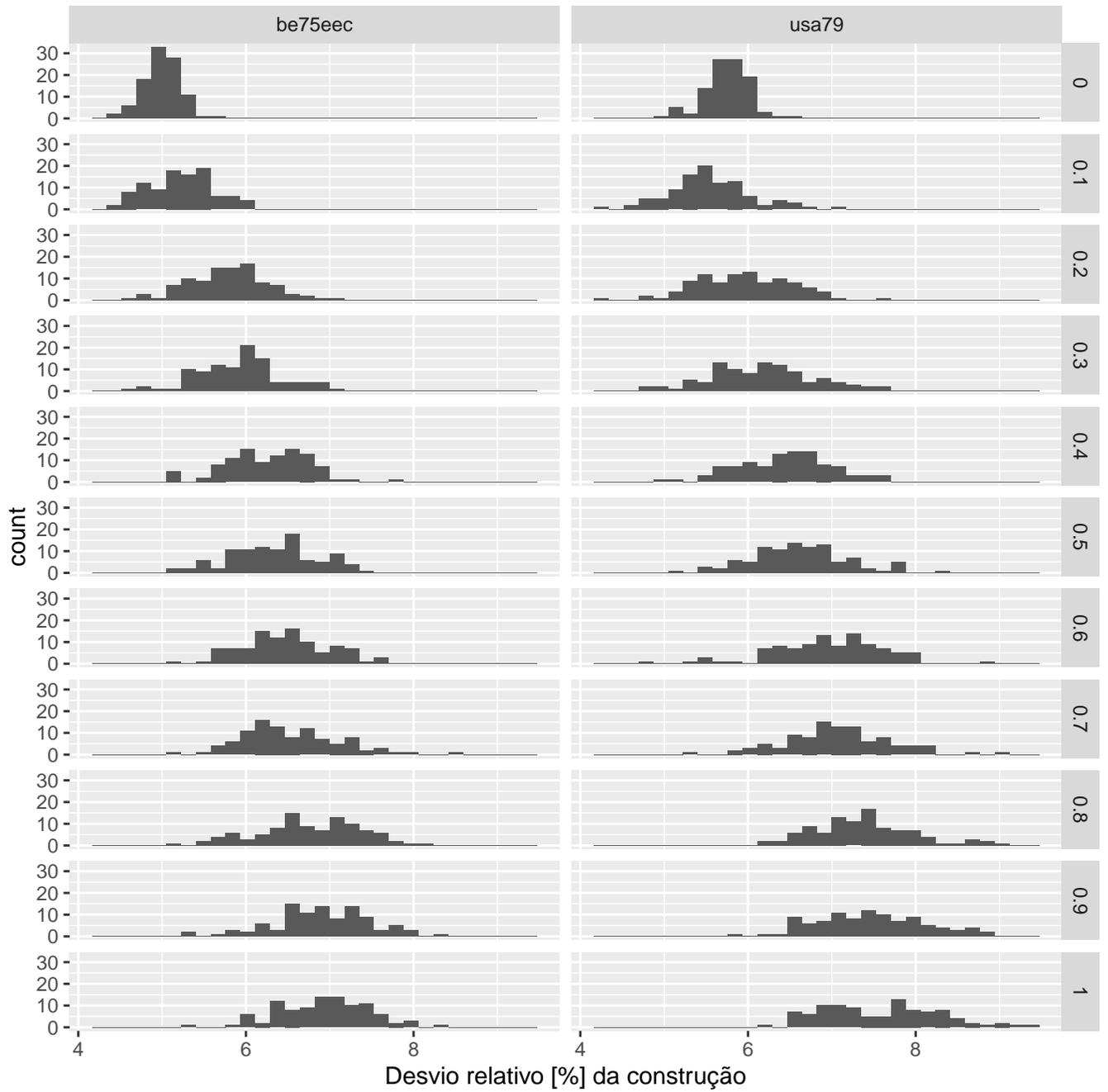


Figura 1: Histogramas para duas instâncias selecionadas.