

## Lista de exercícios 2

### Exercício 1 (Tempera Simulada para o PCV, 5pt)

Estende a busca local 2-opt para o PCV da primeira lista para uma busca com Tempera Simulada. Continua com a mesma vizinhança. Para determinar a temperatura inicial e final ideal, usa a técnica de Johnson et al. (1989) (também explicada na seção 2.3.3 da notas):

- Numa fase inicial, determina a temperatura inicial. Começando com uma temperatura baixa  $T := T_0 = 0.01$ , executa por  $L_i = L/10$  iterações. Caso mais que 70% dos movimentos (i.e. 35 movimentos) foram aceitos, a temperatura atual é aceita. Senão, repete com  $T := 2T$ .
  - Para terminar, mantém um contador quantas das últimas fases com  $L$  movimentos aceitaram menos que 5% dos movimentos. Caso o incumbente melhora, o contador é zerado. Caso ele chegue em 5, termina.
- a) Determina o melhor valor de  $L = 5 \times 2^k$ ,  $k \in [5]$ . Para um determinado  $k$ , escolhe a taxa de resfriamento  $\alpha = 0.99^k$  (isso mantém o trabalho por nível de temperatura fixo, ver Johnson et al. (1989)). Roda um experimento para todo  $L$  com as instâncias da primeira lista, cada instância com 5 replicações, e escolhe o  $L$  que minimiza o desvio relativo média do melhor valor conhecido.
- b) Determina a complexidade empírica do algoritmo.
- c) Compara os resultados com os resultados da primeira lista.

Entrega dos resultados experimentais:

- Para o item a) arquivo texto com os resultados individuais da forma

L	alpha	instance	rep	v0	time	value
10	0.99	nome-1	1	123456	31.4	88123
20	0.98	nome-2	2	234567	55.9	88123
...						

com “L” o valor de  $L$  usado, “alpha” o valor de  $\alpha$  correspondente, “rep” o número da replicação, “instance” o nome da instância, “v0” o valor da solução inicial, “time” o tempo para executar a Tempera Simulada, e “value” o valor da melhor solução encontrada.

### Exercício 2 (Construção gulosa e GRASP para FL, 5pt)

Projeta uma heurística construtiva e um GRASP para o FL.

- a) Implementa um algoritmo guloso para FL. Começa com a solução com nenhuma facilidade aberta (de custo  $\infty$ ). Repetidamente abre uma facilidade que diminui o custo mais possível, até nenhuma nova facilidade consegue diminuir o custo.
- b) Estende o algoritmo do item a) para um algoritmo guloso randomizado com estratégia “Guloso- $\alpha$ ”, com  $\alpha = 0.2k$ , para  $k \in \{0, \dots, 5\}$  e constrói 1000 soluções para cada  $\alpha$ . Plote seis histogramas que mostram a distribuição das 1000 soluções gulosas para as instâncias 2000-10 e MQ5. Qual  $\alpha$  produz os melhores valores?
- c) Estende a construção para uma heurística GRASP por aplicar uma busca local “primeira melhora” do primeira lista em cada solução construída. Repete a avaliação do item anterior. Qual  $\alpha$  agora produz os melhores valores?
- d) Compara os resultados com os resultados da primeira lista.

Entrega dos resultados experimentais:

- Para o item b) & c) um único arquivo texto com os resultados individuais da forma

```
alpha instance rep time0 v0    time  value
0.4  3000-10  1   17.3 123456 31.4  88123
0.6  3000-10  2   32.2 234567 55.9  88123
...
```

com “alpha” o valor de  $\alpha$  usado, “rep” o número da replicação, “instance” o nome da instância, “time0” o tempo para construir uma solução em segundos, “v0” o valor da solução construída, “time” o tempo para a busca local, e “value” o valor da solução final depois da busca local.

## Referências

Johnson, D. S., C. R. Aragon, L. A. McGeoch e C. Schevon (1989). “Optimization by Simulated Annealing. Part I, Graph Partitioning”. Em: *Oper. Res.* 37, pp. 865–892 (ver p. 1).