

## Lista de exercícios 1

### Exercício 1 (Vizinhanças, 1.5 pt)

Considere o problema de coloração de um grafo não-direcionado  $G = (V, E)$  e suponha que temos  $n = |V|$  cores disponíveis. A vizinhança que atribui uma cor diferente a um vértice é simétrica? Fracamente otimamente conectada? Conectada? Exata? (Um vizinho sempre tem que ficar factível.) Qual o caso da vizinhança que troca a cor de dois vértices adjacentes, nas mesmas condições?

### Exercício 2 (Vizinhanças, 2 pt)

Chandra et al. (1999) mostraram que no caso métrico do PCV temos  $2\text{-opt}(x)/\text{OPT}(x) \leq 4\sqrt{n}$ . Agora considera uma métrica que satisfaz  $d_{ik} \leq \alpha(d_{ij} + d_{jk})$  para uma constante  $\alpha > 1$  e qualquer tripla de cidades  $i, j, k$ .

- Mostra que um caminho  $P = (v_1, v_2, \dots, v_k)$  satisfaz  $d_{v_1, v_k} \leq d(P)\alpha^{k-1}$ .
- Mostra que agora temos  $2\text{-opt}(x)/\text{OPT}(x) \leq 4\alpha^n \sqrt{n}$ . (Dica: define  $E_k = \{(c_i, c_{i+1} \mid d_{i, i+1}\} > 2\alpha^n \text{OPT}(x)/\sqrt{k}$ .)

### Exercício 3 (Busca local para o PCV, 3 pt)

Implemente os algoritmos GSAT e WalkSAT como visto em aula. Encontra em <http://www.cs.ubc.ca/~hoos/SATLIB> as instâncias flat100-1, par8-5-c, flat75-1, flat50-1 e conduza os seguintes experimentos:

- Execute cada um dos algoritmos 15 vezes em cada instância. Use para o período nos dois casos  $3n$ .
- Determine para cada execução o tempo até encontrar uma atribuição que satisfaz a fórmula, o número de períodos, e o número de iterações.
- Relate os resultados numa tabela de forma  $\mu \pm \sigma$  para as três medidas e para cada par de algoritmo e instância.
- Compare a complexidade empírica do WalkSAT com a complexidade teórica.

Observações:

- Os tempos relatados em Hoos e Stützle para resolver essas instâncias com uma versão melhorada do GSAT ficam entre 7000 e 70000 passos (e entre 0.09 e 0.39 segundos). Uma implementação simples de teste precisa entre 25000 e 1500000 passos (em até 6 segundos). O WalkSAT puro precisa mais tempo. Em função da eficiência da implementação um tempo limite de alguns minutos parece adequado.
- É importante para o funcionamento do GSAT escolher um dos melhores vizinhos uniformemente. Uma maneira de implementar isso é usar *reservoir sampling*: ao encontrar o  $i$ -ésimo vizinho com o mesmo valor que o melhor vizinho atual, aceitá-lo com probabilidade  $1/i$ . Além disso a busca continua, mesmo num mínimo local.

Entrega dos resultados experimentais:

- Um arquivo texto com os resultados individuais da forma

| alg     | instance  | rep | t   | p | i     |
|---------|-----------|-----|-----|---|-------|
| gsat    | flat100-1 | 1   | 101 | 5 | 3000  |
| walksat | par8-5-c- | 1   | 50  | 3 | 23000 |
| ...     |           |     |     |   |       |

com “alg” o algoritmo, “instance” o nome da instância, “rep” o número da replicação, “t” o tempo para satisfazer a fórmula em segundos, “p” o número de períodos para satisfazer a fórmula, e “i” o número de iterações até satisfazer a fórmula. Caso uma execução não encontrou uma solução que satisfaz a fórmula, as colunas “t”, “p”, e “i” devem informar o valor “NA”. O arquivo deve ter 120 linhas.

#### Exercício 4 (Busca local para QBP, 3.5 pt)

Estuda a busca local no problema de programação quadrática binária (ingl. quadratic binary programming, QBP). Usa as instâncias `bqp{50,100,250,500}`-1 disponíveis na biblioteca Biq Mac (<http://biqmac.uni-klu.ac.at/biqmaclib.html>). Os valores das soluções ótimas são disponíveis em <http://biqmac.uni-klu.ac.at/biqmaclib.pdf>.

- Compare as estratégias “first improvement” e “best improvement”. Repete o experimento 15 vezes, com instâncias iniciais aleatórias.
- Compare com uma busca local monótona randomizada. Testa valores de  $p \in \{0.0, 0.05, 0.25, 0.5, 0.75, 1.0\}$ . Repete o experimento para cada valor de  $p$  cinco vezes. Relata o tempo até encontrar a solução ótima, ou o melhor valor encontrada depois de um tempo limite de 5 min e o número de iterações numa tabela da forma  $\mu \pm \sigma$ .
- Determine a complexidade empírica do algoritmo (o tempo para encontrar a solução ótima em função do tamanho da instância).

Observações:

- Para um desempenho razoável é necessário implementar um cálculo eficiente da função objetivo (ver observação 2.1 nas notas).
- Para escolher um vizinho melhor ou arbitrário uniformemente sem armazenar todos vizinhos pode-se usar *reservoir sampling*: ao encontrar o  $i$ -ésimo vizinho candidato, aceitá-lo com probabilidade  $1/i$ .

Entrega dos resultados:

- Para o primeiro item um arquivo texto com os resultados individuais da forma

```
alg instance    rep time iterations    value
PM  bqp50      1   17.3 23764          88123
PM  bqp50      2   17.3 23764          88123
MM  bqp50      1   32.2 41234          88123
...
```

com o tempo em segundos.

- Para o segundo item um arquivo texto com os resultados individuais, da forma

```
p  instance    rep time iterations value
0.0 bqp50      1   30.1 2943           88123
0.0 bqp50      2   33.8 1234           88124
...
```

com o tempo em segundos. A coluna “rep” é o número de replicação do algoritmo na mesma instância.

**Data de entrega:** 04/10/2017.

## Regras para listas de exercícios

1. Os exercícios podem ser resolvidos em colaboração com outros, mas a entrega é individual informando os eventuais colaboradores.
2. A entrega consiste das respostas (formato PDF, não escrito a mão), o código para questões de implementação e os resultados experimentais detalhadas, caso tem experimentos.
3. Para receber pontos as respostas devem ser justificadas (i.e. provadas quando não são óbvias).
4. Somente entregem respostas que vocês sabem explicar pessoalmente.
5. O limite para as respostas é 6 páginas (sem referências) com parâmetros razoáveis (tamanho do fonte, margens).