

Lista de exercícios 1

Exercício 1 (Vizinhanças, 1pt)

A vizinhança 2-flip para o k -SAT é simétrica? Fracamente otimamente conectada? Exata? E a vizinhança k -flip para $k > 2$?

Exercício 2 (PCV, 2pt)

Mostre que na prova do teorema 2.1 a rota ótima é única.

Exercício 3 (Busca local para um problema polinomial, 4pt)

Implemente a busca local do exemplo da árvore geradora mínima. Testa a busca em grafos aleatórios conexos. (Para gerar um grafo aleatório conexo gera n vértices e insere cada aresta possível com probabilidade $p > 2 \ln n/n$; caso o grafo não é conexo, repete.) Gere 10 grafos com $n = 100$, $p = 0.1, 0.2, 0.3$, com $n = 250$, $p = 0.05, 0.1, 0.2$ e com $n = 500$, $p = 0.05, 0.1, 0.2$. Cada aresta recebe um peso aleatório em $[1, 100]$. Gere uma solução inicial por uma busca em profundidade a partir do primeiro vértice (inclui as arestas “forward” na árvore) e aplica a busca local. Compare as estratégias “primeira melhora” e “melhor melhora” em tempo e número de iterações. Determine a complexidade empírica do algoritmo e compare com o estado de arte.

Observações:

- a) Use uma implementação com listas de adjacência.
- b) O grafo é não-direcionado. Porém uma maneira adequada para manter a árvore geradora é de forma enraizada. Com isso o caminho na árvore entre dois vértices u e v é composto pelo caminho u para l e v para l com l sendo o ancestral comum mais próximo.
- Tarefa opcional (pode ser computacionalmente custoso, 2pt extra): Determina para cada instância o número mínimo e máximo possível de iterações da solução inicial para um mínimo local)

Exercício 4 (Busca local para um problema NP-completo, 3pt)

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 conduz os seguintes experimentos:

- a) Execute cada um dos algoritmos 15 vezes em cada instância. Usa para o período nos dois casos $3n$.
- b) Determine para cada execução o tempo até encontrar uma atribuição que satisfaz a fórmula, e o número de períodos.
- c) Relate os resultados numa tabela de forma $\mu \pm \sigma$ para as duas medidas junto com o número de instâncias resolvidas para cada par de algoritmo e instância.
- d) Compare a complexidade empírica (usando a regressão linear e um modelo exponencial) 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
gsat    flat100-1      1      101      5
walksat  par8-5-c      1      50      3
...
```

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. Caso uma execução não encontrou uma solução que satisfaz a fórmula, as colunas “t”, e “p” devem informar o valor “NA”. O arquivo deve ter 120 linhas.

Data de entrega: 15/09/2025.

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 é eletrônica, não escrito a mão, em formato PDF.
3. Para receber pontos as respostas devem ser justificadas (i.e. provadas quando não são óbvias).
4. Somente entreguem respostas que vocês sabem explicar pessoalmente.