

Trabalho IV

Resolvendo um Problema NP-Completo: Caixeiro Viajante

O objetivo do quarto trabalho da disciplina da Complexidade de Algoritmos será resolver instâncias do problema do caixeiro viajante utilizando três abordagens algorítmicas distintas: *backtracking*, algoritmo guloso e um algoritmo de aproximação baseada em desigualdade de triângulos (vista em aula). Adicionalmente, como bônus, os alunos podem produzir ainda uma solução utilizando programação dinâmica utilizando *bitmask* (uma forma de representação compacta da solução parcial).

Os algoritmos deverão ser executados sobre as entradas fornecidas, as quais variam entre 8 e 10.000 vértices/cidades. Para os algoritmos que demorarem mais do que 10 minutos para terminar uma execução, os alunos deverão fazer uma projeção do tempo que seria necessário para que o algoritmo terminasse o processamento com base nas entradas que foram computadas dentro do tempo limite (considerando a complexidade do algoritmo, isto é, seu crescimento assintótico).

Ao executar os algoritmos sobre as entradas geradas armazene o tempo necessário para a execução e o menor custo obtido, de modo que seja possível comparar o tempo de execução dos diferentes algoritmos e sua eficácia em encontrar soluções boas. Gere gráficos que realizem comparações entre o tempo de execução e o desempenho do algoritmo. Também deverão ser elencadas as complexidades de tempo e espaço dos algoritmos e de suas suboperações.

Linguagens permitidas: C, C++, Java e Python.

Material a ser apresentado:

- Códigos das implementações;
- Gráfico comparativo de tempo de execução para as abordagens;
- Análise de complexidade de tempo e espaço para o melhor e pior caso de ambas as abordagens e suas suboperações;

Restrições:

- Este trabalho poderá ser realizado no máximo em duplas
- Deverá ser apresentado no máximo até o dia **05/12/2018 (sob agendamento)**
- Perguntas serão realizadas para validar a autoria dos códigos e resultados