Teoria dos Grafos – COS 242

2012/2

Trabalho de Curso – Parte 3

1 Logística

Esta é a terceira parte do trabalho de curso. Você deve realizar esta parte utilizando e estendendo a biblioteca implementada nas duas outras partes. Se você fez a primeira parte em dupla, então a dupla deve continuar a mesma. Seu relatório deve informar as decisões de projeto e implementação do algoritmo para resolver o problema abaixo e deve conter no máximo 4 páginas. Além disso, você deve responder às perguntas relacionadas aos estudos de caso. O trabalho será apresentado em aula, onde você irá explicar as funcionalidades e fazer uma rápida exposição dos estudos de caso (10 minutos para apresentação).

2 Descrição – Parte 3

Neste trabalho, você deve projetar e implementar um algoritmo para resolver o clássico problema do *caixeiro viajante* (ou *TSP - Travelling Salesman Problem*, em inglês). No problema do caixeiro viajante, temos um conjunto de cidades e um custo associado em ir diretamente de uma cidade para outra. O problema é encontrar um caminho de menor custo que parte de uma cidade qualquer, passa por todas as outras cidades exatamente uma vez, e retorna à cidade inicial.

O problema do caixeiro viajante pode ser modelado utilizando um grafo completo com n vértices. Neste grafo, cada vértice corresponde a uma cidade e cada aresta corresponde a possibilidade de ir diretamente de uma cidade para outra. Desta forma, pesos são associados às arestas do grafo, correspondendo aos custos em ir diretamente de uma cidade para outra. A figura abaixo ilustra o problema com 5 cidades onde o percurso de menor custo é dado por $1, 2, 3, 4, 5, 1$.

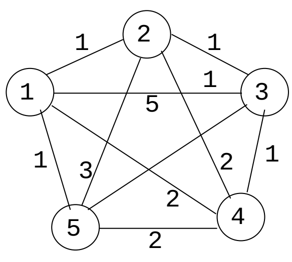

Figura 1: Exemplo do problema do caixeiro viajante com cinco cidades

Na sua forma mais geral, os custos associados ao translado de uma cidade a outra (isto é, os pesos das arestas) podem assumir qualquer valor. Entretanto, existem diferentes variações do problema, que o torna mais específico, mas não necessariamente mais fácil. Uma dessas é a variação Euclideana, onde cidades correspondem a pontos no plano e custos correspondem a distâncias Euclideanas entre os pontos. Neste caso, cada cidade c_i é representada por um par ordenado (x_i, y_i) que corresponde à sua localização no plano.

Você deve projetar e implementar um algoritmo para resolver o problema do caixeiro viajante em sua variação Euclideana. É importante ressaltar que não se conhece solução eficiente (ou seja, tempo de execução polinomial) para este problema, nem mesmo para sua variação Euclideana. Desta forma, você deve projetar algoritmos que não necessariamente irão retornar a solução ótima, ou seja, o percurso de menor custo, mas que retornem soluções *próximas* ao ótimo. Por exemplo, você pode projetar um algoritmo guloso para o problema, mas que não necessariamente irá obter o percurso ótimo em todas as instâncias.

Para a variação Euclideana do problema, você pode explorar o fato de que os custos (ou seja, as distâncias) respeitam a *inequação do triângulo*. Ou seja, para quaisquer três pontos (vértices), a, b, c, temos que $d(a, c) \leq d(a, b) + d(b, c)$, onde $d(p, q)$ é o custo (distância) entre os pontos p e q.

Sua biblioteca deve ser capaz de receber como entrada um conjunto de n pontos no plano que irão representar os vértices do grafo. Neste caso, a primeira linha informa o número de pontos no plano e cada linha subsequente informa as coordenadas (x, y) de cada ponto, conforme ilustrado abaixo. Sua biblioteca deve então construir o grafo completo correspondente, onde cada ponto no plano corresponde a um vértice e o peso de cada aresta é dado pela distância Euclideana entre os vértices incidentes à aresta.

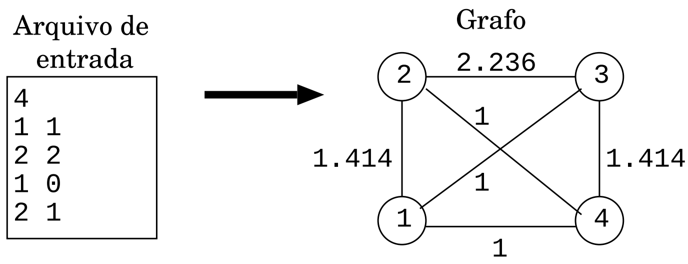

Figura 2: Formato de entrada via pontos no plano e grafo correspondente.

3 Estudos de Caso

Calcule o percurso de menor custo para todos os grafos disponíveis no website da disciplina. Para cada grafo, seu algoritmo deve informar o custo total do percurso obtido, assim como o percurso em si (ou seja, a sequência de vértices que o determina). Prepare uma tabela com as distâncias dos percursos para cada um dos grafos do estudo de caso.

Por fim, utilize o website da disciplina para visualizar o percurso encontrado por seu algoritmo!