Universidade Federal de Lavras

Trabalho de Grafos

Franciscone Luiz de Almeida Júnior

Igor Emanuel Carvalho Cruz

Introdução

A Teoria dos Grafos é um ramo da matemática que estuda as relações entre objetos de um determinado conjunto. Esse conceito teve como primeiro trabalho o artigo de LeonhardEuler sobre as sete pontes de Königsberg, o que ilustra a aplicação de um problema real da teoria de grafos. Atualmente a teoria dos grafos tem aplicabilidade variada, podendo ir desde a obter ou gerar algum dado em um caminho entre cidades (Google Maps, exemplo), até obter dados para estrategias de marketing de pessoas que possuem alguma conexão nas redes sociais.

No contexto desse trabalho a Teoria dos Grafos é utilizada para resolver problemas que envolvem caminhos entre cidades e de entre pontos turísticos. Esse problemas possuem certas peculiaridades que devem ser consideradas, os quais serão descritos nos tópicos a seguir.

Descrição do problema

O problema consiste em ajudar um estudante, Pedro, a encontrar a melhor rota em seu *“tour”* pela Europa mesmo possuindo sérias restrições orçamentais. No decorrer do problema vemos que Pedro possui um limite de tempo por dia e ao final deste tempo deve retornar ao hostel mais próximo para descanso. Os objetivos para solucionar o problema são minimizar o numero de dias viajado e a duração total da viajem.

Algoritmos Implementados

Com base no Problema do Caixeiro Viajante o algoritmo escolhido para resolução do problema foi o Vizinho mais Próximo. Após fazer a leitura do arquivo de texto, na primeira linha contém número de hostels, número de pontos e o tempo de permanência em cada hostel. Ao final da leitura das coordenadas dos hostels inicia-se a coleta das coordenadas dos pontos turísticos e do tempo que sera gasto em cada um. O armazenamento dos pontos foram feitos em um vetor e a incidência de ida em cada um deles é salvo em um vector.

Algorítimo do Vizinho mais Próximo

Pseudocódigo

01. Inicia em um vértice arbitrário como vértice atual.
02. Encontre o vértice adjacente (não visitado) ao vértice atual com a menor distância.
03. Marca o vértice atual como visitado.
04. Torna o vértice encontrado como atual.
05. Se todos os vértice do grafo estão marcados como visitados:
    5.1. volte ao vértice inicial e termina execução.
    5.2. senão volte ao passo 02.

A execução

A função é iniciada verificando que existem hostels e a partir disso são setados o tempo_atual, o tempo_gasto e a distância com o valor zero. Na seguinte etapa começa um “*loop*” que, enquanto existem vértices não visitados será repetido, “*loop*” este que consiste em verificar o vértice mais próximo da localização atual, onde o primeiro ponto é atribuído como o hostel lido na posição da metade do tamanho do vetor de hostels (número de hostels : 2), incrementando o tempo_gasto e o tempo_total.

Como exite um tempo limite de permanência em cada vértice, se o tempo_atual for maior que o tempo de permanência no vértices é procurado o hostel mais próximo, se não procura-se o novo vértice(ponto turístico) mais próximo,em seguida o vértice da condição satisfeita se torna o novo vértice atual, incrementando o tempo_atual em relação a distância já percorrida. Toda vez que se seleciona um novo vértice como atual, é salvo em um vector de ancestrais o vértice de partida, para que se possa ter um bom controle do trajeto que se foi percorrido.

Ao final da execução do programa é impresso em tela um valor lógico que indica se o problema dado tem solução ou não (0 ou para falso e 1 para verdadeiro) e o número de viagens necessárias para fazer o percurso e o tempo total gasto para a viagem.

Resultados

Melhor solução encontrada

Nome	Pts. tur.	bkv viagens	bkv tempo
P1	100	9	9595,6
P2	100	3	9560
P3	48	2	1412,2
P4	192	5	4217,4
P5	40	4	3817,4
P6	40	4	842,9
P7	40	2	1160,5
P8	40	2	1259,5

Tabela 01: Soluções ótimas para os problemas.

Onde:

Nome: problema testado;

Pts. tur.: número de pontos turísticos;

bkv viagens: melhor valor conhecido de número de viagens;

bkv tempo: melhor valor conhecido de tempo total gasto.

Solução encontrado pelo algoritmo do trabalho

Instância	Sol.	t(s)	GapViagens(%)	GapTempo(%)
P1	9	9960,15	0,00	3,80
P2	3	9775,38	0,00	92,21
P3	2	1706,97	0,00	20,87
P4	3	2774,02	-40,00	-34,22
P5	4	3927,68	0,00	2,89
P6	6	1373,56	50,00	62,96
P7	2	1585,14	0,00	36,59
P8	2	1665,75	0,00	32,25

Tabela 02: Soluções obtidas pelo algoritmo do trabalho.

Onde:

Instância: problema testado;

Sol: valor da solução obtida (número de viagens realizadas);

t(s): tempo gasto pelo tour;

GapViagens: Relação percentual entre o número de viagens obtido na melhor solução e o resultado encontrado pelo algoritmo do trabalho;

GapTempo: Relação percentual entre o tempo de viagem obtido na melhor solução e o tempo obtido pelo algoritmo do trabalho.

Conclusão

O problema do caixeiro viajante apesar de um ser um problema extremamente complexo, possuindo vários algoritmos mais complexos para solucionar considerados NP-difíceis conseguindo garantir ate 1,5 da solução ótima, optamos por uma solução heurística, no caso do Vizinho mais Próximo, mesmo que este não alcance uma solução ótima alcança soluções plausíveis conforme exposto na tabela de resultados encontrado podendo-se notar que na maioria dos casos o problema chega a ser solucionado com valores relativamente próximos a solução ótima.