

Trabalho Pratico

Problema do Caixeiro Viajante (PCV)

AEDS II

Definição

O Problema do Caixeiro Viajante (Travelling Salesman Problem - TSP¹) tenta determinar a menor rota para percorrer um conjunto de cidades, retornando sempre à cidade de origem.

Exemplo 1

Dado o conjunto de cidades: São Paulo, Rio de Janeiro, Belo Horizonte e Brasília. E as distâncias entre elas (ver Tabela 1²). Encontrar o menor percorrido saindo e chegando na mesma cidade.

Cidades	São Paulo	Rio de Janeiro	Belo Horizonte	Brasília
São Paulo	0	436	587	1022
Rio de Janeiro	436	0	442	1166
Belo Horizonte	587	442	0	735
Brasília	1022	1166	735	0

Tabela 1. Distâncias em quilômetros entre as cidades.

O menor caminho é fazer a rota:

Brasília – Belo Horizonte – Rio de Janeiro - São Paulo – Brasília

Total de quilômetros percorridos: **2630**

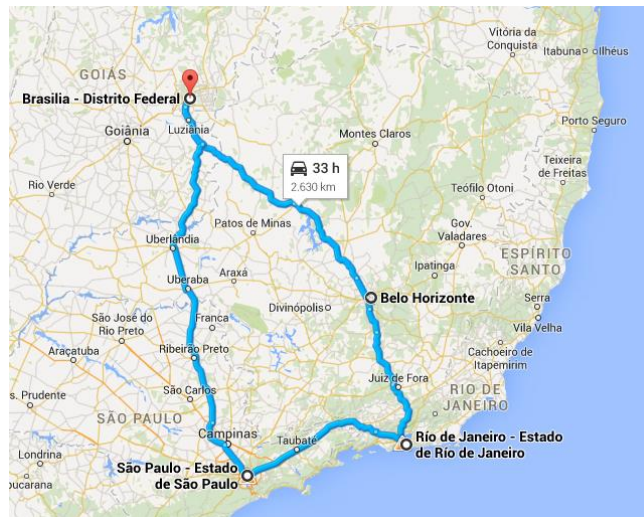


Figura 1. Mapa do menor percorrido entre 4 cidades brasileiras.³

¹ http://en.wikipedia.org/wiki/Travelling_salesman_problem

² Dados obtidos do site: <http://www.entrecidadesdistancia.com.br/>

³ Imagem obtida do google maps (<https://goo.gl/maps/Bzz4y>)

O **Problema do Caixeiro Viajante Euclidiano** é um caso particular do **PCV** onde as distancias entre as cidades estão dadas por pontos num plano e obedecem a desigualdade triangular.

A distância euclidiana entre dois pontos A e B está dada por:

$$d_{AB} = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

Dados três pontos no espaço A, B e C, a desigualdade triangular está dada por:

$$d_{AB} \leq d_{AC} + d_{CB}$$

Exemplo 2

Substituindo as distâncias do exemplo 1 por coordenadas temos:

Cidades	Coordenadas	
	X	Y
São Paulo (SP)	2	0
Rio de Janeiro (RJ)	5	1
Belo Horizonte (BH)	4	4
Brasília (BSB)	1	10

Tabela 2. Coordenadas das cidades em duas dimensões.

O tour (caminho) ótimo do exemplo anterior cumpre a desigualdade triangular:

BSB – BH – RJ – SP - BSB

A soma total das distâncias para este tour está dada por:

$$d_{T_1} = d_{BSB-BH} + d_{BH-RJ} + d_{RJ-SP} + d_{SP-BSB} \approx 22,08$$

No processo de formação de cada tour deve-se comprovar que três cidades cumprem a desigualdade triangular. Para o tour BSB – BH – RJ:

$$d_{BSB-RJ} \leq d_{BSB-BH} + d_{BH-RJ}$$

$$9,85 \leq 9,86$$

Outro possível tour pode ser: BH – SP – RJ – BH – BSB - BH

$$d_{T_2} \approx 23,03$$

[5pts] Parte I

Criar um TAD⁴ city e um TAD tour e implementar a solução para o PCV Euclidiano que imprime todas as soluções viáveis ou tours possíveis. Dois tours com o mesmo percorrido mas com cidades de origem diferentes serão considerados diferentes.

Formato

As coordenadas das cidades são número inteiros (tipo Int).

Arquivo de entrada

A primeira fileira do arquivo de entrada tem o número de cidades (N). Na primeira coluna vão as coordenadas X das cidades e na segunda coluna as coordenadas Y. Numa fileira somente estarão as coordenadas X e Y de uma cidade, separadas por **TAB**.

N

a_x a_y

b_x b_y

c_x c_y

...

Arquivo de saída

A primeira fileira do arquivo de saída tem o número de soluções/tours viáveis (N). Cada fileira vai conter uma solução/tour viável do PCV. As cidades que formam parte do tour estão separadas por **TAB**. As coordenadas X e Y de cada cidade devem estar dentro de parêntesis e separadas por vírgula.

N

(a_x, a_y) (b_x, b_y) (c_x, c_y) ...

(b_x, b_y) (a_x, a_y) (c_x, c_y) ...

(c_x, c_y) (a_x, a_y) (b_x, b_y) ...

...

[5pts] Parte II

Deve encontrar a solução ou soluções ótimas do PCV Euclidiano para os seguintes números de cidades:

- 4
- 6
- 8

⁴ Tipo Abstracto de Dado

- 10
- 11
- 12
- 13
- 14

Inclua também o tempo de execução do seu algoritmo para cada uma das instâncias anteriores. Caso não consiga encontrar a solução para alguma destas instancias deve justificar o porquê. Observe que pode ter várias soluções ótimas como no **exemplo 1** onde tem 4 tours ótimos com o mesmo percorrido mas cada um saindo de uma cidade diferente.

[3pts] Parte III (Pontos Extras!!)

Deve solucionar o PCV usando a **heurística do vizinho mais próximo** (Nearest Neighbor), que consiste em iniciar o tour em uma cidade e continuar com a cidade mais próxima ainda não visitada. O formato dos arquivos de entrada e saída será o mesmo da **parte I**.

Também deve calcular a complexidade desta heurística e fazer uma tabela de comparação entre a solução da heurística e a solução ótima com as oito instâncias da **parte II**. A comparação vá a ser feita da seguinte maneira calculando os *gaps* (G) entre a solução ótima S^* e a solução da heurística S :

$$G = \frac{S - S^*}{S^*}$$

Exemplo 3

Para o conjunto de 4 cidades apresentado no **exemplo 2** o tour ótimo tem uma distância total de 22,08 e um outro tour que não é o ótimo tem uma distância total de 23,03. A comparação entre estas duas soluções é feita da seguinte maneira:

$$G = \frac{23,03 - 22,08}{22,08} \approx 0,04$$

O valor de 0,04 é o *gap* entre as duas soluções do PCV.

Entrega:

- 1. Comece a fazer este trabalho imediatamente. Você nunca terá tanto tempo para resolvê-lo quanto agora!**
- 2. Data de entrega: 01 de Outubro de 2017, até às 23:59 horas, ou antes.**
- 3. Submissão: Envie este trabalho para o email tiagocunha87@gmail.com (OBS1: talvez até a data da entrega eu consiga acesso ao moodle.) (OBS2: devido aos vários problemas com o envio do tp0, recomendo fortemente que vocês coloquem o arquivo no google drive ou dropbox e me enviem o link.) tendo como assunto seu nome e como anexo um arquivo comprimido, descrito abaixo, com o nome onde o string "SeuNomeCompleto" é o seu nome completo sem espaços em branco.**

Exemplo para o aluno José das Couves:

Assunto: AEDS2 TP1: José das Couves

Arquivo comprimido: AEDS2_TP1_José_das_couves

contendo apenas os seguintes arquivos:

- (a) Os arquivos main.c, caixeiro.c e caixeiro.h: arquivos fontes na linguagem C.**
- (b) O arquivo documentacao.pdf: documentação**

- 4. Entrada: Você vai receber o nome do arquivo com a instância do problema via linha de comando**
- 5. Saída: A saída do seu algoritmo será escrita em 3 arquivos com os seguintes nomes: (por exemplo para instância com 4 cidades) instancia_4_q1.txt, instancia_4_q2.txt e instancia_4_q3.txt**
- 6. Linguagem: Você deve escrever o seu programa obrigatoriamente na linguagem de programação C. Lembre-se de não utilizar bibliotecas ou funções específicas do Windows.**
- 7. Documentação: Documentação sobre o trabalho (em .pdf), incluindo as decisões tomadas relativas a cada parte do tp e detalhes de especificação que porventura estejam omissos no enunciado. A documentação deve conter entre outras coisas:**
 - 1. Introdução: descrição sucinta do problema a ser resolvido e visão geral sobre o funcionamento do programa.**
 - 2. Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados.**
 - 3. Estudo de Complexidade: estudo da complexidade do tempo de execução dos procedimentos implementados e do programa como um todo (notação O).**