Problema do caixeiro viajante - TSP

O problema do caixeiro-viajante (PCV), ou como é conhecido em inglês: travelling salesman problem (TSP), é um problema de otimização que, apesar de parecer modesto é, na realidade, um dos mais estudados, por cientistas, matemáticos e investigadores de logística, genética e produção, entre outros (Applegate et al., cop. 2006, p. 1). O problema pertence à categoria NP-hard (em inglês), que o remete para um campo de complexidade exponencial, isto é, o esforço computacional necessário para a sua resolução cresce exponencialmente com o tamanho do problema. Assim, dado que é difícil, se não impossível, determinar a solução ótima desta classe de problemas, os métodos de resolução passam pelas heurísticas e afins que, do ponto de vista matemático, não asseguram a obtenção de uma solução ótima (Cunha, 2002).

Definição e formulação do problema

Em termos leigos, imaginem um território de vendas (por exemplo, o Estado do Paraná), que tem diversas cidades polo que devem ser visitadas pelo vendedor. Ele começa e termina na sua base e deve visitar uma e só uma vez cada cidade polo. O critério para a escolha da ordem das cidades é minimizar a distância total percorrida.

Usando uma definição acadêmica:

> O problema do caixeiro-viajante consiste na procura de um circuito que possua a menor distância, começando numa qualquer cidade, entre várias, visitando cada cidade precisamente uma vez e regressando à cidade inicial (Nilsson, 1982).

O tamanho do espaço de procura aumenta exponencialmente (fatorialmente) dependendo da quantidade de cidades n, o número de cidades, uma vez que existem $(n-1)!/2$ circuitos possíveis (a posição inicial é arbitrária, e a ordem do circuito pode ser invertida).

A solução do PCV pode ser de dois tipos: a exata (impraticável em instância médias ou grandes, devido ao comportamento exponencial) e as aproximadas, todas dependentes de alguma heurística.

Começando pelo segundo grupo, há diversos métodos. Estes são procedimentos particulares, o que os torna inflexíveis para a determinação de boas soluções para um outro problema ligeiramente diferente.

Quadro esquemático de soluções aproximadas

- construção de circuítos (alg. guloso)
- árvore de cobertura mínima
- *simulated annealing*
- algoritmo de busca tabu
- redes neurais
- algoritmos genéticos
- *aint colony optimization algorithm*

Solução Exata

Embora demorada esta é a solução que vai ser trabalhada neste exercício. É um bom exemplo de algoritmo recursivo (neste caso, claramente a recursão vem em nosso auxílio: este mesmo algoritmo não recursivo teria enorme complicação e dificuldade) e suficiente pequeno para poder rodar em minutos usando um computador pessoal simples.

Um caso real

Em 1998 uma equipe de matemáticos encontrou o caminho mais curto para visitar as 13.509 cidades americanas que tinham, naquele ano, mais de 500 habitantes. Foram necessários 3,5 meses de processamento de três multiprocessadores (32 pentium cada) ligados em rede. Aqui, eliminaram-se as rotas obviamente ineficientes logo de cara. O problema: a estratégia só vale para este problema e para estas cidades.

Outro caso real

Os algoritmos mostrados nesta folha e rodados em um micro bem lerdinho (um pentium 4 com 2.8GHz, com 632 MB de RAM, mas rodando Linux e neste sob WMware um Windows XP, e neste rodando APL2 da IBM) e em um micro rápido (2 CPUS de 2.80GHz, 3.3GB de memória e com APL2 sob Windows XP nativo)

qtd de cidades	CPU lerda	CPU rápida
6 cidades	31 milisseg	desprezível
7 cidades	156 milisseg	78 milisseg
8 cidades	1.1 seg	500 milisseg
9 cidades	9 seg	4 seg
10 cidades	81 seg	37 seg
11 cidades	836 seg	367 seg
12 cidades	153 min	67 min
13 cidades	30.6 horas	13.4 horas
14 cidades	16.5 dias	7.2 dias

O problema que você vai resolver

Há uma lista de 6 cidades paranaenses, cada uma com suas coordenadas obtidas a partir de um eixo cartesiano imaginário montado sobre o paralelo 26^o SUL (x) e sobre o meridiano 54^oO (y). No cálculo, considerou-se uma projeção plana do território, o que certamente introduzirá pequenas distorções. Não importa, avante ! Seja um exemplo de 6 cidades:

```
1-CANDIDO ABREU       269 157
2-GEN CARNEIRO        268 -49
3-GUAIRA              -25 213
4-JACAREZINHO         399 315
5-PALMAS              200 -54
6-PARANAGUA           543  51
```

Supondo que a cidade inicial seja 3 (Guaira), a sequência de cidades a visitar é 3 1 4 6 2 5 3 e a distância é de 1514.57 km

Algoritmos

Para construir a matriz de distâncias entre pares de cidades, o algoritmo é

```
funcao ACHADIST (COORD: nome,xx,yy[nn]) : matd [nn;nn]
i,j : inteiro
para i = 1 to nn
    para j = 1 to nn
        matd[i][j] ← COORD[i;2 3] distancia COORD[j;2 3]
    fim{para}
fim{para}
```

Note-se que a função `distancia` acima usada é a simples aplicação do Teorema de Pitágoras sobre o mapa. Note-se também que se o TSP deve ser feito sobre outro conceito que não a distância linear, esta função deve ser substituida pela que calcula o método desejado. Este pode ser: distância via estradas, tempo de voo, custo do pedágio, ...

Eis a matriz de distâncias para o exemplo

	Cand Abr	Gen Car	Guaira	Jacarez	Palmas	Pgua
	--------	-------	-------	-------	-------	-------
Cand Ab	.000	206.002	299.286	204.607	221.995	293.789
Gen Car	206.002	.000	393.056	386.855	68.184	292.617
Guaira	299.286	393.056	.000	436.096	349.162	590.650
Jacarez	204.607	386.855	436.096	.000	419.240	300.719
Palmas	221.995	68.184	349.162	419.240	.000	358.712
Pgua	293.789	292.617	590.650	300.719	358.712	.000

TSP e o algoritmo guloso

Se uma resposta degradada for aceitável, pode-se aplicar o algoritmo guloso neste problema. A estratégia aqui é *das cidades não visitadas, devo escolher a mais próxima.* Em um caso real, obtiveram-se os seguintes valores:
Cidades: 1-Cascavel, 2-Fco Beltrao, 3-Guaira, 4-Guarapuava, 5-Irati, 6-Ortigueira, 7-Paranavai, 8-Ponta Grossa, 9-Telêmaco Borba e 10-Umuarama.
Inicial: Ponta Grossa:
Solução exata: 1111.97 Km. Sequência: 8 5 4 2 1 3 10 7 6 9 8. Algoritmo Guloso: 1329.56 Km. Sequência: 8 5 4 6 9 7 10 3 1 2 8

☞ Para você fazer

Suponha a seguinte lista de cidades

```
1-ADRIANOPOLIS        495 146
2-IRATI               338  62
3-MARINGA             211 287
4-PARANAVAI           158 323
5-TELEM BORBA         341 185
6-UMUARAMA             72 246
```

Sendo que o início e o fim do ciclo de viagens deve ocorrer na cidade de PARANAVAI . Após calcular o menor caminho entre todas as cidades, responda:

1. Qual o comprimento em quilometros do menor caminho ?
2. Qual o comprimento em quilometros do caminho guloso ?

m.c.	guloso

Note que como a instância deste problema é pequena é perfeitamente possível que os valores (caminho mínimo e caminho guloso) sejam iguais.