QXD0037 - Inteligência Artificial

Trabalho de Implementação I – Especificação
Subida de Encosta p/ o Caxeiro Viajante
Prof. Samy Sá, 2016.2

PUBLICAÇÃO: 05/10/2016
PRAZO: 21/10/2016

Neste projeto, você deverá utilizar a técnica de Subida de Encosta para atacar o problema do Caxeiro Viajante. Este documento determinará o formato de entrada e saída que seu programa deve seguir e detalha a representação de estados e operadores que devem ser implementados.

Obs.: Partes do código servirão para trabalhos futuros envolvendo o mesmo problema.

O Problema do Caixeiro Viajante (TSP)

Enunciado: Dado um conjunto de cidades, as distâncias entre cada par destas cidades, e uma cidade inicial, encontre o menor caminho que passa exatamente uma vez por cada cidade e retorna à cidade inicial.

Para simplificar o problema, suponha que todas as cidades são conectadas. Desta forma, o mapa das cidades deve ser considerado um grafo completo em que os nós são as cidades e cada aresta representa o caminho entre as cidades em suas extremidades.

Representação do Problema

Para aplicar a Subida de Encosta ao TSP, considere que:

- As cidades são pontos com coordenadas ***x***, ***y*** em um mapa (um plano).
- A distância entre quaisquer duas cidades é dada diretamente pela distância geométrica entre os pontos do plano em que as cidades estão situadas. Ou seja, dadas duas cidades com coordenadas (x_1,y_1) e (x_2,y_2), a distância entre as cidades será dada por $\text{sqrt}((x_1 - x_2)^2 + (y_1 - y_2)^2)$.
- Os estados do espaço de busca serão permutações na lista de cidades existentes. Em um problema com 6 cidades A,B,C,D,E,F, por exemplo, a sequência [B,D,A,F,E,C] indica o circuito que começa na cidade B, segue para D, depois para A, etc., até C e então retorna a B. Uma vez que esta sequência representa um circuito nas cidades listadas, a representação de um estado deve ser percebida como uma lista circular.
- A função heurística a ser utilizada é o custo total do circuito. Como temos um problema de minimização, um valor menor de heurística será considerado melhor.

O estado inicial e os operadores a serem trabalhados serão especificados em variações da implementação.

Formatos de entrada e saída

A entrada do programa será um arquivo com coordenadas de pontos. O arquivo terá duas linhas: a primeira terá todas as coordenadas ***x*** de cada cidade no plano e a segunda terá todas as coordenadas ***y*** de cada cidade no plano. Por exemplo, a entrada do programa pode conter o conteúdo:

```
0.0  1.0  2.0  0.0  1.0  1.0  3.0
0.0  2.0  0.0  2.0  1.0  3.0  2.0
```

O conteúdo acima indica uma entrada com 7 cidades em que as coordenadas da primeira são ***x*** = 0.0

e ***y*** = 0.0, as coordenadas da segunda cidade são ***x*** = 1.0 e ***y*** = 2.0, e assim por diante.
Para facilitar os cálculos de custos de circuitos no seu programa, utilize a entrada para calcular uma matriz M[i,j] das distâncias entre cada par de cidades i, j.

Para a saída do seu programa base, exiba na tela, a cada iteração do algoritmo de Subida de Encosta, a sequência de visita das cidades no estado atual e o custo total deste circuito.

Variações da Implementação

Utilizaremos dois critérios diferentes para o estado inicial e dois tipos de operadores para a vizinhança entre os estados. Você terá, então, 4 variações de implementação que deverão ser comparadas, mas a diferença de implementação entre estas variações é pequena e gerarão pouco trabalho a mais.

Estado Inicial 1: Utilize a sequência em que as cidades são fornecidas no arquivo de entrada.

Estado Inicial 2: Gere uma permutação aleatória das cidades fornecidas.

Operador 1: Trocar duas cidades quaisquer de lugar. Neste operador, você deve escolher duas cidades quaisquer e trocar as duas de lugar na sequência. Por exemplo, se você tiver 6 cidades A,B,C,D,E,F e aplicar este operador para trocar as cidades D e E de lugar no estado [B,D,A,F,E,C], obterá o estado [B,E,A,F,D,C]. Você pode então calcular o custo do novo circuito e comparar ao original para determinar se este é ou não melhor.

Operador 2: Inverter trechos das permutações. Neste operador, você deve escolher duas cidades quaisquer e inverter toda a sequência que começa na primeira e vai até a segunda. No nosso exemplo com 6 cidades A,B,C,D,E,F, se aplicarmos este operador para trocar as cidades D e E de lugar no estado [B,D,A,F,E,C], obterá o estado [B,E,F,A,D,C], ou seja, todo o trecho que inicia na cidade D e vai até a cidade E será invertido. Você pode então calcular o custo do novo circuito e comparar ao original para determinar se este é ou não melhor.

Randomização da vizinhança: Ao gerar a vizinhança de um estado, aleatorizar a sequência em que seus vizinhos serão avaliados pela subida de encosta.

Observe que a ordem em que as cidades são escolhidas para aplicar o Operador 1 não faz diferença no estado resultante, mas faz diferença na aplicação do Operador 2. Considere um exemplo com 7 cidades A,B,C,D,E,F,G e o estado [B,E,F,A,D,C,G]. Se utilizarmos o Operador 1 para trocar as cidades F,C (índices 3 e 6 do vetor) de lugar, obteremos [B,E,C,A,D,F,G], o mesmo resultado obtido se o Operador 1 receber como parâmetros as cidades C,F (índices 6 e 3 do vetor). Com o Operador 2, o comportamento é outro. Se utilizarmos o Operador 2 para trocar as cidades F,C (índices 3 e 6 do vetor) de lugar, obteremos [B,E,C,D,A,F,G], mas se o Operador 2 receber como parâmetros as cidades C,F (índices 6 e 3 do vetor), o resultado será [B,G,CA,D,F,E], pois estaremos invertendo o trecho "C,G,B,E,F", invés do trecho "F,A,D,C".

Os critérios acima permitirão executar 8 variedades da Subida de Encosta sobre o TSP:

1. Estado Inicial 1 com Operador 1 ***sem*** randomização da vizinhança.
2. Estado Inicial 1 com Operador 1 ***com*** randomização da vizinhança.
3. Estado Inicial 1 com Operador 2 ***sem*** randomização da vizinhança.
4. Estado Inicial 1 com Operador 2 ***com*** randomização da vizinhança.
5. Estado Inicial 2 com Operador 1 ***sem*** randomização da vizinhança.
6. Estado Inicial 2 com Operador 1 ***com*** randomização da vizinhança.
7. Estado Inicial 2 com Operador 2 ***sem*** randomização da vizinhança.
8. Estado Inicial 2 com Operador 2 ***com*** randomização da vizinhança.

Comparação Entre as Variações

Dado um arquivo de entrada, seu código final deve executar cada variação acima 20 vezes. Você pode colocar os códigos das 8 variações em um loop único e executar as iterações dos métodos de maneira alternada. Produza um arquivo único tabulando os custos de melhor caminho encontrados em cada simulação. A tabela deve conter uma coluna para cada variação da técnica (8 colunas) e uma linha para cada repetição na simulação (20 linhas). Você pode usar os números de 1 a 8 para nomear as colunas conforme o código na seção anterior.

Em um relatório curto (1-2 páginas), descreva suas observações no experimento proposto. Compare os resultados obtidos por você apontando quais parâmetros favoreceriam os melhores resultados.

É encorajado que outras variações sejam propostas e realizadas para comparação. Se for o caso, descreva as demais variações no seu relatório, bem como os efeitos destas variações nas suas conclusões.