UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS

Modelagem e Resolução de Problemas aplicando conhecimentos de Inteligência Artificial

Disciplina: Inteligência Artificial **Prof.:** Bonfim Amaro Junior

Data da Entrega: 05/10/2020 Semestre : 2020.1

1. Objetivo

O auxílio de mecanismos mais eficientes de "Inteligência Computacional" na tentativa de encontrar soluções para um problema são de fundamental importância para o contexto da inteligência artificial. O objetivo desse trabalho é investigar métodos de busca e formatos de representação para resolução de um problema específico: N-caixeiros viajantes.

2. Problema: *M*-CAIXEIROS VIAJANTES

O Problema do Caixeiro Viajante (PCV) é um problema que tenta determinar a menor rota para percorrer uma série de cidades (visitando uma única vez cada uma delas), retornando à cidade de origem. É um problema de otimização NP-difícil inspirado na necessidade dos vendedores em realizar entregas em diversos locais (as cidades) percorrendo o menor caminho possível, reduzindo o tempo necessário para a viagem e os possíveis custos com transporte e combustível.

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

Exemplificando, consideremos 6 cidades com as distâncias dadas pela tabela a seguir:

Cidade	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	5
5	9	7	8	2	0	2
6	1	2	6	5	2	0

Uma possível solução para o exemplo considerado é s = (1 4 2 5 3 6).Para esta solução, a distância total percorrida é *dist* = d(1,4) + d(4,2) + d(2,5) + d(5,3) + d(3,6) + d(6,1) = 4 + 9 + 7 + 8 + 6 + 1 = 35. Observe que qualquer permutação das n cidades representa uma solução para o PCV. O que queremos é, dentre todas as possíveis permutações (soluções), determinar aquela cuja distância total percorrida é a menor possível.

Se d(i,j) = d(j,i) diz-se que o PCV é simétrico. Caso contrário ele é dito assimétrico.

Para o PCV simétrico há (n-1)! / 2 soluções possíveis. Para mostrar a magnitude do espaço de soluções, para n = 20 (número de cidades), e considerando como 0,00000001s, o tempo de cálculo de um computador para uma rota, seriam necessários 19 anos para encontrar a melhor solução a partir da enumeração de todas as possibilidades.

Já para o problema dos *m*-Caixeiros Viajantes há *m* Caixeiros e se deseja minimizar a distância total (Euclidiana) percorrida por todos eles. Considere os seguintes parâmetros de entrada:

$Cidades$: Conjunto das cidades
d_{ij} : Distância da cidade i à cidade j
n : Cardinalidade do conjunto de cidades, isto é, $n = |Cidades|$
m : Número de Caixeiros Viajantes

(a) Variáveis de decisão:
x_{ij} variável binária que assume valor 1 se o arco (i, j) for utilizado e 0, caso contrário
f_{ij} quantidade de fluxo enviada da cidade i para a cidade j

(b) Função objetivo:
$$\min \sum_{i \in Cidades} \sum_{j \in Cidades} d_{ij} x_{ij}$$

(c) Restrições:

c.1) À cada cidade k, exceto a origem (cidade de índice 1), só chega um arco:
$$\sum_{i \in Cidades} x_{ik} = 1 \;\; \forall k \in Cidades \,|\, k \neq 1$$

c.2) De cada cidade k, exceto a origem (cidade de índice 1), só sai um arco:
$$\sum_{j \in Cidades} x_{kj} = 1 \;\; \forall k \in Cidades \,|\, k \neq 1$$

c.3) Da cidade origem saem m arcos:
$$\sum_{j \in Cidades} x_{1j} = m$$

c.4) À cidade origem chegam m arcos:
$$\sum_{i \in Cidades} x_{i1} = m$$

c.5) Exceto para a cidade origem (primeira cidade), o fluxo que chega a uma cidade k menos o que sai de k é igual a 1:
$$\sum_{i \in Cidades} f_{ik} - \sum_{j \in Cidades} f_{kj} = 1 \;\; \forall k \in Cidades \mid k \neq 1$$

c.6) O fluxo máximo que passa em um arco usado no percurso é inferior a $n - m$, onde n é o número de cidades e m é o número de caixeiros:
$$f_{ij} \leq (n-m)x_{ij} \;\; \forall i \in Cidades, \;\; \forall j \in Cidades$$

c.7) Integralidade e não-negatividade:
$$x_{ij} \in \{0,1\} \;\; \forall i \in Cidades, \;\; \forall j \in Cidades$$
$$f_{ij} \geq 0 \;\; \forall i \in Cidades, \;\; \forall j \in Cidades$$

3. Desenvolvimento

A Equipe, contendo no máximo 4 participantes, deve implementar, computacionalmente, em qualquer linguagem de programação, uma heurística ou metaheurística adaptada ao problema dos *m*-caixeiros viajantes descritos na seção 2.

O parâmetro *m* (número de caixeiros) deve ser testado para 1 <= *m* <= 3.

Os arquivos ncit*.dat são anexados com a seguinte configuração: na primeira linha é exibido o número de cidades, e nas linhas seguintes, as coordenadas (x,y) de cada cidade. A ordem de apresentação das coordenadas das cidades nos arquivos ncit*.dat é arbitrária. Percorrendo as cidades nesta ordem, a distância resultante é maior que a distância mínima possível. Não existe nenhuma relação direta entre os dois problemas. Por exemplo, as 30 cidades presentes em ncit30.dat não representam um subconjunto das 100 cidades de ncit100.dat.

O objetivo é encontrar uma ordenação de tal modo (considerando *m*-caixeiros) que, uma vez percorrendo as cidades nesta ordem, a distância resultante seja mínima. Não esquecer de contabilizar também o caminho de retorno da última cidade para a primeira.

Obs1: Para validar sua função de custo, com a ordenação atual de cada cidade nos arquivos, a distância para percorrer as cidades, na sequência, apresenta os seguintes valores (apenas um caixeiro):

- 30 cidades: 123865,3550
- 100 cidades: 109952,5821

Obs2: As distâncias ótimas conhecidas para os dois casos são dadas a seguir:

- 30 cidades: 48872,4026
- 100 cidades: 109952,5821

Estas soluções ótimas devem servir apenas para indicar a quão próxima do ótimo se encontra uma solução fornecida pelo seu algoritmo. Ou seja, estes valores ótimos não podem estar presentes no código de seu programa, exceto para guiar sua de alguma forma a busca pela solução.

Assim para medir a performance do seu algoritmo, vamos considerar que cada caixeiro só pode rodar 600km por dia e que sua diária custa 150 reais.

Além da implementação, a equipe deve executar o programa 10 vezes para cada instância: ncit30.dat e ncit100.dat. Os resultados devem ser apresentados no formato de tabela com as seguintes informações:

Instância: ncit30

Exec.	*m*	Distância Encontrada	Distância de Cada Caixeiro	Quant Dias	Custo Total	% do ótimo
1	1	49250,450	49250,450	82	12.300	*
2	2	50500,780	1-> 30025,080 2-> 20475,700	50 34	7.500+ 5.100 = 12.600	

* Colocar o cálculo que informa qual a proximidade (em porcentagem da sol.ótima para cada cidade descritas na **Obs2**.

4. Avaliação

O trabalho vale de 0 à 10 e deve ser apresentado pela equipe diretamente ao professor em reunião virtual pela ferramenta meet no link: https://meet.google.com/gri-jwax-wji. Serão destinados dois dias para apresentação (plano de ensino): 06 e 08 de outubro de 2020. Esse trabalho será contabilizado como substituição da nota da AV2.