



FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA
ENSINANDO E APRENDENDO

TRABALHO DE IMPLEMENTAÇÃO COMPUTACIONAL
PESQUISA OPERACIONAL
CIÊNCIAS DA COMPUTAÇÃO

OBJETIVO: IMPLEMENTAR A UTILIZAÇÃO DE UMA METAHEURÍSTICA COMPUTACIONAL (ALGORÍTMOS GENÉTICOS) APLICADA AO PROBLEMA DO CAIXEIRO VIAJANTE (TRAVELING SALESMAN PROBLEM - TSP).

CONTEÚDO DE APRENDIZADO : MÉTODOS COMPUTACIONAIS (METAHEURÍSTICOS) E OTIMIZAÇÃO COMBINATÓRIA.

AVALIAÇÃO:

- Entendimento do código
- Padrão de desenvolvimento
- Apresentação dos Resultados
- Apresentação oral (apenas para o professor) do desenvolvimento.

Obs: Qualquer componente da equipe poderá ser solicitado.

DATA DE ENTREGA :

- 17/11/2017 (VALENDO 11,0 PONTOS) **(TUDO IMPLEMENTADO)**

- **24/11/2017 (NOTA MÁXIMA 10,0) (PRAZO FINAL)**

FUNDAMENTAÇÃO

O problema do caixeiro viajante é um problema muito famoso, muito estudado e que possui natureza combinatória. Um caixeiro viajante, partindo de sua cidade, deve visitar exatamente uma única vez cada cidade de uma dada lista e retornar para casa tal que a distância total percorrida seja a menor possível. Este problema tem inúmeras aplicações práticas, como minimização de rotas de veículos, confecção de sistemas digitais, sequenciamento de atividades e outros.

Para testar todas possibilidades de rotas para N cidades seria necessário calcular a permutação $N!$. Tudo bem, para um rota com 3,4 ou 5 cidades, é fácil de calcular. Agora imagine um tour com 30 cidades! Seria necessário calcular a distância total de cada uma dessas 2.65×10^{32} diferentes rotas, assumindo um trilhão de cálculos por segundo, o que levaria aproximadamente 252.333.390.232.297 anos para achar a menor rota. Adicione mais uma cidade a esta rota e o tempo aumentaria em um fator exponencial elevado a 31. Fica claro, que é uma solução impossível calculando manualmente ou por métodos exatos de modelagem em programação linear inteira.

Algoritmos genéticos podem ser usados para solucionar este problema em menor tempo. Embora não haja garantias de encontrar a melhor solução, ela pode achar uma solução satisfatória para um tour de 100 cidades em um “curto” período de tempo.

Há alguns passos básicos para solucionar o problema do caixeiro viajante usando algoritmos genéticos, vejamos cada um deles a seguir:

1. Um conjunto aleatório de rotas são criadas, inicializando assim a população. Este algoritmo cria uma população inicial 'gulosa', o qual dá preferência em montar conexões entre as cidades que estão mais próximas uma da outra.

2. Escolha as 2 melhores rotas (menores) da população para o cruzamento (crossover) e combine-os para gerar 2 novas rotas filhas. Grandes chances destas novas rotas filhas serem melhores que os seus pais.
3. Em raros casos, as rotas filhas sofrem o processo de mutação. Isto é feito para prevenir que todas as rotas da população sejam idênticas
4. As novas rotas filhas são inseridas na população substituindo no lugar de 2 rotas das maiores existentes. O tamanho da população se mantém constante.
5. As novas rotas filhas são repetidamente criadas até atingirem um critério de parada desejado (melhor rota).

REQUISITOS

Cada cromossomo representará uma possível rota (Versão Simétrica (Distância de Ida = a da Volta)). Assim, as soluções deverão satisfazer duas restrições:

1. Cada cidade a ser visitada apareça em alguma rota;
2. A mesma cidade não pode aparecer mais de uma vez;

Cada equipe deverá:

- Implementar a representação do cromossomo como formato de uma solução.
- A função de Aptidão (Fitness) (Função Objetivo) Custos de movimentação entre cada par de cidades da rota, calculando o custo entre a última e a primeira (VOLTA). Use a **distância Euclidiana** entre dois pontos.

- Implementar os componentes do AG : CROSSOVER, MUTAÇÃO, SELEÇÃO E CRITÉRIO DE PARADA.

EXECUÇÃO

Os parâmetros utilizados serão :

Tamanho da População : 80 indivíduos

Taxa de Crossover : 0.75 - (75%)

Taxa de Mutação : 0.1 – (10%)

Critério de Parada: Se a melhor solução não ficar ainda melhor para as próximas 40 gerações, pare.

Método de Seleção da ROLETA.

Método de Criação da Nova população: ELITISMO.

Observações:

1. A equipe deverá ler o arquivo 30CIT.TXT com a seguinte configuração: na primeira linha é apresentado o número de cidades, e nas linhas seguintes, as coordenadas (x,y) de cada cidade. A ordem de apresentação das coordenadas das cidades no arquivo é arbitrária. Percorrendo as cidades nesta ordem, a distância resultante é maior que a distância mínima possível.
2. O objetivo é encontrar uma ordenação de tal modo que, uma vez percorrendo as cidades nesta ordem, a distância resultante seja mínima.
3. Não se esquecer de contabilizar também o caminho de retorno da última cidade para a primeira.
4. Com a ordenação atual, a distância para percorrer as cidades, na sequência, apresenta os seguintes valores:
 - 30 cidades: 123865,3550
5. A distância ótima conhecida para o arquivo é:

- 30 cidades: 48872,4026

6. Esta solução ótima deve servir apenas para indicar o quão próxima do ótimo se encontra uma solução fornecida pelo seu algoritmo. De forma alguma estes valores ótimos devem ser utilizados para obter o fitness de um candidato a solução ou direcionar de algum modo o processo de evolução. Ou seja, estes valores ótimos não podem estar presentes no código de seu programa.

Resultado Final apresentar a configuração da melhor rota, o valor do fitness (Solução) e qual a porcentagem da sua solução perante a “ótima” supracitada.