

Exercício de A.C.O (Ant Colony Optimization)

Para exercitar a aplicação dos conceitos de A.C.O., estamos propondo o seguinte exercício:

Criar uma implementação de A.C.O. para o problema de detecção do menor caminho. O objetivo é conseguir o maior número de convergências possíveis para o menor caminho em 10 execuções.

1. Implementação

A implementação pode ser feita de duas formas:

- a) Modificando o algoritmo A.C.O. que está implementado no programa de visualização;
- ou
- b) Criando uma solução própria que produza o arquivo de resultados para ser verificado no programa de visualização;

Uma restrição, é que o número de formigas, nunca deve exceder o total de cidades e o número de ciclos é limitado a 1000. Sem essa limitação, a implementação seria quase força bruta e o ACO não teria sentido.

Fica como sugestão de implementação, nos dois casos, uma das variações do algoritmo original de A.C.O.:

- Ant Colony System - ACS
- MAX-MIN Ant System - MMAS
- Elitist Ant System
- Rank-based Ant System
- Best-Worst Ant System

1.1 Modificando o algoritmo de A.C.O. do Programa de Visualização

O código fonte que foi entregue com este enunciado deverá ser modificado para permitir uma nova implementação do algoritmo de A.C.O.

1.2 Criando uma Solução Própria

Para quem desejar criar sua própria solução, os seguintes requisitos devem ser atendidos:

- a) A solução deve ser capaz de carregar os exemplos produzidos pelo programa de visualização (como os próprios arquivos de exemplos disponibilizados);
- b) A solução deve executar por 10 vezes para produzir um arquivo de resultados;
- c) A solução deve exportar esse arquivo de resultados em formato pré-definido, para que os mesmos possam ser carregados no programa de visualização.

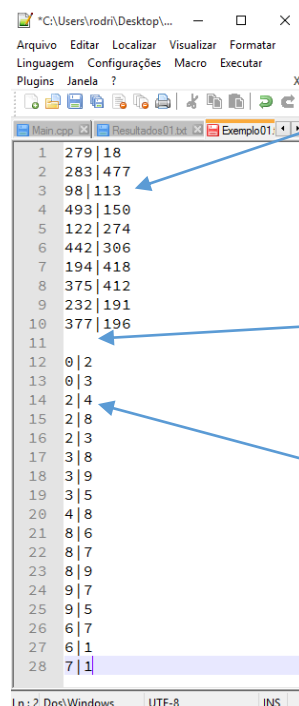
- d) A solução pode, de forma adicional, exportar um determinado cenário produzido por ela (cidades e caminhos) exatamente no mesmo formato dos arquivos de exemplo, para que, desse modo, esse cenário possa ser carregado na solução disponibilizada pelo grupo de ACO.

2. O Programa de Visualização

O programa de visualização implementa a versão original do algoritmo de A.C.O de Dorigo, porém com algumas modificações intencionais. Além disso, o mesmo possui as seguintes funcionalidades:

- Criação do Grafo de cidades / caminhos;
- Visualização do Grafo de cidades / caminhos;
- Exportação de arquivo com o grafo de cidades / caminhos (permitindo que quem desenvolver outra solução, possa gerar o grafo na solução disponibilizada e exportar em um arquivo para poder manipular o mesmo);
- Visualização dos Resultados;
- Importação de arquivo com grafo de cidades / caminhos (permitindo carregar os exemplos disponibilizados ou permitindo que os grupos que desenvolveram outra solução, possam exportar seu cenário para poder ser visualizado e manipulado na solução disponibilizada)
- Importação de arquivo de resultados. Esse arquivo deve possuir 11 linhas (uma com os parâmetros utilizados e uma para cada execução do algoritmo de ACO);

2.1 Exemplo do Arquivo de Configuração (para quem desenvolver outra solução)



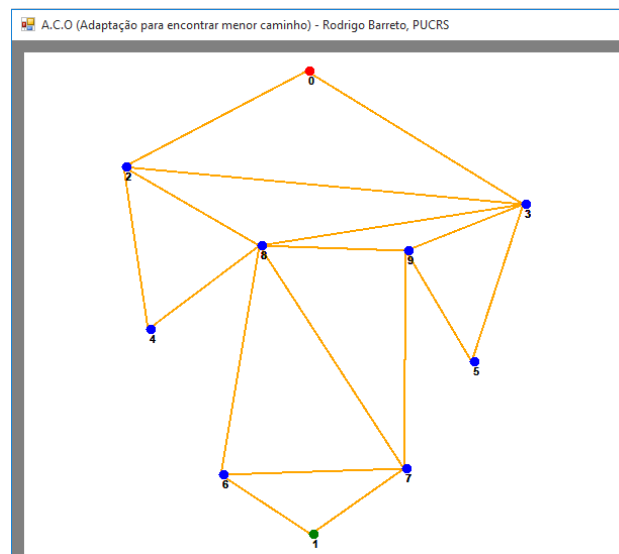
```
1 279|18
2 283|477
3 98|113
4 493|150
5 122|274
6 442|306
7 194|418
8 375|412
9 232|191
10 377|196
11
12 0|2
13 0|3
14 2|4
15 2|8
16 2|3
17 3|8
18 3|9
19 3|5
20 4|8
21 8|6
22 8|7
23 8|9
24 9|7
25 9|5
26 6|7
27 6|1
28 7|1
```

Pares de coordenadas x e y de cada ponto (cidade). Cada linha equivale a uma cidade. A primeira linha é a origem, a segunda linha é o destino, e as outras linhas são as outras cidades

Uma linha em branco para separar o conjunto de pares de cidades do conjunto de pares de caminhos

Pares de pontos (cidades) que define uma aresta (caminho). De | Para

Esse exemplo de arquivo, modela essa configuração:

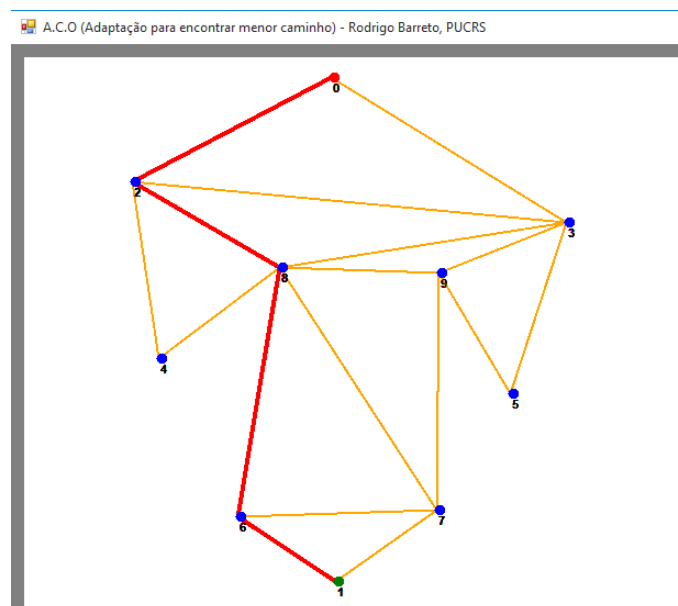


2.2 Exemplo do Arquivo de Exportação de Resultados (para quem desenvolver outra solução)

O arquivo de resultados deverá ser um arquivo texto, contendo 11 linhas. A primeira linha contém os parâmetros *Alfa*, *Beta*, *Rho*, Número de Formigas e Número de Iterações. As linhas restantes deverão ser cada um dos 10 resultados. Como separador de cada informação, deve ser usado o caracter "|". As linhas de resultado nada mais são que o Número das Cidades na ordem em que são percorridas.

```
12.5|2.9|0.2|100|1000
0|2|8|6|1
```

Essa linha de resultado dá origem a essa solução:



A solução gerada usando um α de 12.5, um β de 2.9, um ρ de 0.2, com 100 formigas em 1000 iterações/passos).

3. Como será feita a validação do exercício

No dia da apresentação dos grupos, serão utilizados 3 arquivos de exemplos que não serão disponibilizados previamente. Eles contem respectivamente 40, 60 e 80 cidades.

Quem modificou a solução disponibilizada, carregará cada um destes exemplos e deverá executar o ACO (clicando no botão “EXECUTA ACO”) 10 vezes.

Quem criou outra solução, deverá:

- Carregar na sua solução os nossos arquivos de exemplo;
- Executar sua solução 10 vezes, gerando um arquivo de resultados (conforme seção 2.2);
- Importar no programa de visualização o arquivo de resultados gerado.

A avaliação do exercício será feita baseada na apresentação das modificações realizadas pelo grupo, a variabilidade das soluções encontradas de convergência para o melhor caminho e a explicação das diferenças da solução desenvolvida para a solução disponibilizada.

4. Referências

- A.Coloni, M.Dorigo, V.Maniezzo, "Distributed Optimization by Ant Colonies," Proceedings of the First European Conference on Artificial Life, Paris, France, F.Varela and P.Bourguin (Eds.), Elsevier Publishing, 134–142, 1991.
- Dorigo M, Gambardella LM: **Ant colony system: a cooperative learning approach to the traveling salesman problem**. *IEEE Transactions on Evolutionary Computation* 1997, 1:53–66.
- Stützle T, Hoos H: **Improvements on the Ant-System: Introducing the MAX-MIN Ant System**. In *Artificial Neural Nets and Genetic Algorithms*. Springer Vienna; 1998:245–249.
- **A new rank based version of the Ant System. A computational study**. [<http://epub.wu.ac.at/616/>]
- Dorigo M, Maniezzo V, Coloni A: **Ant system: optimization by a colony of cooperating agents**. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 1996, 26:29–41.
- Zhang Y, Wang H, Zhang Y, Chen Y: **BEST-WORST Ant System**. In *2011 3rd International Conference on Advanced Computer Control (ICACC)*; 2011:392–395.