

Trabalho Prático 2
Ant Colony Optimization

Data de Entrega: 05 de Novembro de 2018

1 Introdução

O objetivo desse trabalho é desenvolver conceitos chaves para construções de soluções para problemas usando Ant Colony Optimization (ACO), envolvendo o entendimento e a implementação dos componentes básicos de um arcabouço de ACO, bem como a análise de sensibilidade dos seus parâmetros (como eles afetam o resultado final, a natureza de convergência, etc) e procedimentos para avaliação das soluções alcançadas. Para esse trabalho, vocês devem elaborar soluções para o problema conhecido como *longest path problem*¹. Dado um grafo $G(V, E)$, uma função $w : E \rightarrow \mathbb{R}$ que atribui pesos a cada aresta e dois vértices $u, v \in V$, denotaremos como \mathcal{P} o conjunto de caminhos **simples**² partindo de u e chegando em v . O problema consiste então em encontrar $P^* = \{e_1^*, e_2^*, \dots, e_k^*\}$ tal que

$$P^* = \arg \max_{P \in \mathcal{P}} \sum_{e_i \in P} w(e_i)$$

Ou seja, queremos encontrar o caminho simples de u a v que maximize o peso total do caminho.

Decisões de Implementação:

1. Como representar uma solução;
2. Como calcular a *fitness* de uma solução;
3. Como calcular a probabilidade de transição entre vértices;
4. Como lidar com soluções inválidas.

2 Bases de Dados

Nesse trabalho, vocês deverão utilizar os três grafos fornecidos como entrada. O primeiro grafo possui 100 vértices e 8020 arestas. Já o segundo grafo possui 20 vértices e 190 arestas. Por fim, o terceiro grafo possui 1000 vértices e 499500 arestas. Todos os grafos são direcionados e ponderados, com pesos inteiros entre 1 e 10. Os vértices estão numerados de 1 à n e cada linha do arquivo corresponde a uma aresta do grafo, fornecendo em ordem o número do vértice do qual ela sai, o número do vértice no qual ela chega e o peso da aresta.

¹https://en.wikipedia.org/wiki/Longest_path_problem

²Um caminho simples é tal que não passa pelo mesmo vértice mais de uma vez.

Para os dois primeiros grafos, sabemos a solução ótima. Para o primeiro grafo, a solução ótima é 990. Já para o segundo grafo, a solução ótima possui peso 168. Vocês devem usar esses valores como referência para avaliar a solução proposta para o trabalho.

3 Metodologia Experimental

O ACO deve ser testado nas bases de dados descritas acima, procurando o caminho mais longo entre o primeiro vértice (de número 1) e o último vértice (de número n). A avaliação experimental deve ser realizada para cada base de dados e inclui a análise de sensibilidade dos principais parâmetros do ACO (e de qualquer decisão de implementação não-usual) e a discussão dos resultados obtidos. Abaixo, temos uma lista não-exaustiva de parâmetros/decisões de implementação a se avaliar:

- Número de formigas;
- Número de iterações;
- Taxa de evaporação;
- Seu algoritmo gera soluções inválidas? Se sim, qual o impacto das possíveis maneiras de lidar com esse problema?

Lembrem-se que o ACO é um algoritmo *estocástico*. Logo, é necessário realizar diversas repetições de uma mesma configuração para podermos concluir qualquer coisa. O desvio-padrão das suas soluções é um bom indicador do número necessário de repetições a serem feitas.

4 Entrega

- Código fonte do programa;
- Documentação do trabalho:
 - **Introdução**
 - **Modelagem e Implementação:** descreva claramente o seu modelo, as dificuldades e as decisões de implementação adotadas, incluindo detalhes da representação, *fitness*, etc
 - **Análise Experimental:** análise dos parâmetros/decisões no resultado do ACO e discussões de resultados. Comentários óbvios do tipo “Podemos ver pela Figura 2 que a função a é maior que a função b .” são desnecessários. Apresente conclusões obtidas, facilite o trabalho do leitor: “A Figura 2 mostra que no contexto X a função a é sempre superior. Isso comprova nossa tese de que ...”
 - **Conclusões**
 - **Bibliografia**

A entrega DEVE ser feita pelo Moodle na forma de um único arquivo zipado, contendo o código e a documentação do trabalho.