

## Trabalho Prático 2 Ant Colony Optimization

Data de entrega: 11/11/2015

### 1 Introdução

O objetivo desse trabalho é desenvolver conceitos chaves para construções de soluções para problemas usando *Ant Colony Optimization* (ACO), envolvendo o entedimento 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*<sup>1</sup>. 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*<sup>2</sup> partindo de  $u$  e chegando em  $v$ . O problema consiste então em encontrar  $P^* = \{e_1^*, e_2^*, \dots, e_k^*\}$  tal que

$$P^* = \arg \max_{P \in \mathcal{P}} \sum_{e_i \in P} w(e_i) \quad (1)$$

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

#### Decisões de Implementação:

- Como representar uma solução;
- Como calcular a fitness de uma solução;
- Como calcular a probabilidade de transição entre vértices;
- 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  $N$ .

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.

O formato dos arquivos de entrada segue a seguinte lógica. Na primeira linha, temos um inteiro indicando o número de vértices do grafo. Na segunda linha, temos dois inteiros indicando o vértice de origem e o vértice de destino, respectivamente, para o problema do caminho máximo. Em seguida, temos  $M$  linhas, cada uma associada a uma aresta. Nessas linhas, o primeiro inteiro indica o vértice de origem, o segundo inteiro indica o vértice de destino e o terceiro inteiro indica o peso da aresta.

<sup>1</sup>[https://en.wikipedia.org/wiki/Longest\\_path\\_problem](https://en.wikipedia.org/wiki/Longest_path_problem)

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

### 3 Metodologia Experimental

O ACO deve ser testado nas bases de dados descritas acima. 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 é um bom indicador do número necessário de repetições e intervalos de confiança devem ser utilizados sempre que possível.

### 4 O Que Deve Ser Entregue

- Código-fonte;
- Documentação do trabalho:
  - Introdução;
  - Implementação: descrição sobre implementação do algoritmo, incluindo detalhes de representação, fitness, etc.
  - Experimentos: análise dos parâmetros/decisões no resultado do ACO e discussões de resultados;
  - Conclusões;
  - Bibliografia.