Proposta de projeto: MCMC

Amanda Ferreira de Azevedo[1], **Wanderson Douglas Lomenha Pereira**[1]

[1] Universidade Federal do Rio de Janeiro
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa em Engenharia
Programa de Engenharia de Sistemas e Computação

{afazevedo,wlomenha}@cos.ufrj.br

1. Definição do problema

Seja $G = (V, E)$ um grafo simples, conexo e não orientado, onde V é o conjunto de vértices e E é o conjunto de arestas. Associe um custo não-negativo c_e à cada aresta $e = \{i, j\} \in E$. Denota-se por d_{ij} ao comprimento do menor caminho simples ligando os vértices $i, j \in V$, ou seja, à *distância* entre eles no grafo G. Por fim, o *diâmetro* de G, d, é dado pela maior distância existente entre qualquer par de vértices de G, em termos de número de arestas. Além disso, seja *B* um número positivo que impõe um limite superior para o quanto se pode gastar na escolha das arestas de uma árvore geradora $T = (V_T, E_T)$. Denomina-se por *Budget Minimum Diameter Spanning Tree Problem* (BDSTP) o problema de encontrar uma árvore geradora T tal que a soma total de suas arestas não ultrapasse B e seu diâmetro seja o menor possível. O problema foi proposto por [Plesnik 1981], sob uma denominação imprecisa onde foi identificado como NP-Difícil. Uma ilustração de uma árvore geradora ótima para o problema é dada na Figura 1.

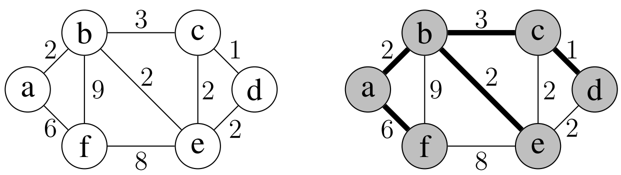

Figura 1. Ilustração de uma árvore geradora de diâmetro mínimo restrita a B = 14. Diâmetro igual a 4.

2. Proposta

Embora ainda pouco investigado na literatura, o BDMSTP é desafiador e têm um grande potencial de aplicações práticas. Em especial, esse problema foi investigado na dissertação de um dos autores deste projeto[1] onde foi implementado os primeiros algoritmos exatos para o problema. No entanto, o problema se mostrou bastante complicado quando tomado valores mais restritos de B. Neste projeto, gostaríamos de construir:

1. Algoritmos de **Monte Carlo** que nos dê soluções de qualidade aliados a tempos de execução pequenos, no intuito de encontrar limites superiores eficientes para serem utilizados em técnicas mais avançadas de otimização.
2. Criação de uma **cadeia de Markov** no intuito de resolver o problema de forma aproximativa utilizando **Simulated Annealing**.

[1] https://www.cos.ufrj.br/index.php/pt-BR/publicacoes-pesquisa/details/15/2974

Para (**1**), pensamos em utilizar a técnica **rejection sampling** para criar árvores geradoras viáveis a partir de árvores geradoras aleatórias melhorarando sua qualidade com uma busca local.

Para (**2**), pensamos em dois caminhos:

- Cada estado da cadeia de Markov será uma solução viável de baixa qualidade (diâmetros grandes) para o problema gerada pelo *algoritmo de Prim*. As transições entre os estados serão construídas a partir de trocas entre vértices. Usaremos *Metropolis-Hasting* para lidar com as restrições de ciclo e de capacidade. A função pegará cada estado e calculará seu diâmetro, priorizando minimizar o diâmetro.
- Cada estado da cadeia de Markov será uma solução inviável (custo maior que o requisitado) de menor diâmetro possível. O cálculo do menor diâmetro possível em uma árvore geradora é um problema *fácil de resolver*, pois se assemelha ao *1-center problem* [Hassin and Tamir 1995]. As transições entre os estados serão construídas a partir de trocas entre vértices. Usaremos *Metropolis-Hasting* para lidar com as restrições de ciclo e de capacidade. A função pegará cada estado e calculará seu custo total, onde priorizaremos minimizar o custo e tornar o problema viável.

Para finalizar, pretendemos fazer uma comparação dessas técnicas com os resultados exatos, analisando tempo e eficiência de cada um.

Referências

Hassin, R. and Tamir, A. (1995). On the minimum diameter spanning tree problem. *Information Processing Letters*, 53(2):109–111.

Plesnik, J. (1981). The complexity of designing a network with minimum diameter. *Networks*, 11(1):77–85.