Iterative Approach for the Minimum Branching Vertices Spanning Tree Problem

Lamiaa EL OUATILI

December 19, 2025

1 Introduction

The *Minimum Branching Vertices Spanning Tree* (MBVST) problem is a combinatorial optimization problem that seeks a spanning tree of a given graph $G = (V, E)$ while minimizing the number of branching vertices (vertices of degree greater than 2). Formally, given a graph G with vertex set V and edge set E, the goal is to find a tree $T = (V, E_T)$ such that $|E_T| = |V| - 1$ and the number of vertices $v \in V$ with $\deg_T(v) > 2$ is minimized.

This report presents an iterative approach using a Linear Programming (LP) formulation, combined with cycle breaking heuristics and component reconnection strategies, suitable for both small and large graphs.

2 LP Formulation

We define the following decision variables:

- $x_e \in \{0, 1\}$ for each edge $e \in E$, where $x_e = 1$ if edge e is included in the tree.
- $y_v \in \{0, 1\}$ for each vertex $v \in V$, where $y_v = 1$ if vertex v is a branching vertex ($\deg(v) > 2$).

The objective is:

$$\min \sum_{v \in V} y_v \tag{1}$$

3 Iterative Heuristic Approach

Since the LP may produce solutions containing cycles or disconnected components, we implement an iterative procedure:

1. **Solve the LP:** Solve the MBVST LP on the full edge set.
2. **Check Connectivity:** If the LP solution is a connected spanning tree, stop.
3. **Reconnect Components:** If disconnected, identify connected components $C_1, \ldots, C_k$ and reconnect them using edges from E, selecting edges that minimize the creation of new branching vertices.

4. **Cycle Breaking:** Identify cycles in the current graph and remove edges to form a tree while preserving connectivity of the main component. Typically, the edge that minimally increases branching is chosen, or randomly if multiple candidates exist.

5. **Iterate:** Use the repaired edge set as input for the next LP iteration. Repeat until the solution is connected or a maximum number of iterations is reached.

Mathematically, the cycle breaking operation can be expressed as:

$$E_{\text{forest}} = E_{\text{connected}} \setminus \{e \in E_{\text{cycle}}\}, \tag{2}$$

where E_{cycle} is an edge set forming a cycle in the current graph.
Reconnection is modeled as:

$$\text{Select } e = (u, v) \in E \text{ such that } u \in C_i, v \in C_j, \text{ minimizing } \nVdash_{\deg(u)>2} + \nVdash_{\deg(v)>2}. \tag{3}$$

4 Results and Visualization

The iterative procedure was implemented in Python using `NetworkX` for graph operations and `PuLP` for LP solving. Each iteration can be visualized, showing:

- Selected edges from the LP.
- Cycles highlighted in red.
- Reconnected components in blue.

This visualization allows the user to inspect how cycles are removed and components are gradually connected, ultimately producing a connected spanning tree with minimal branching vertices.

5 Conclusion

The iterative LP + heuristic approach effectively handles large graphs by:

- Gradually repairing connectivity while controlling branching vertices.
- Visualizing intermediate solutions to debug and improve heuristics.