Connecting the University of Havana:
Design and Analysis of a Degree-Constrained Fiber Optic Network

José Agustín Del Toro González

1 Introduction

The modernization of network infrastructure is a fundamental requirement for the academic and scientific development of modern universities. In this context, the University of Havana, with technical support from ETECSA, aims to interconnect its main buildings through a high-speed fiber optic network.

This work addresses the problem from the perspective of *Design and Analysis of Algorithms*, modeling it as a combinatorial optimization problem on graphs subject to real-world technical constraints. The goal is to design a minimum-cost network that connects all buildings, avoids unnecessary cycles, and respects the limited number of ports available at each network device.

2 Informal Problem Description

A set of university buildings must be interconnected using fiber optic links. Each possible connection has an associated installation cost, and each building is equipped with a network device that supports only a limited number of direct connections.

The objective is to select a subset of connections that:

- Connects all buildings.
- Contains no cycles.
- Respects degree constraints at each building.
- Minimizes the total installation cost.

3 Mathematical Formalization

3.1 Graph Model

The problem is modeled as an undirected weighted graph:

$$G = (V, E, w)$$

where:

- V represents the set of buildings.
- E represents the set of possible fiber connections.
- $w : E \to \mathbb{R}^+$ assigns a cost to each connection.

Each vertex $v \in V$ is associated with a degree bound $b(v)$ representing the maximum number of allowed connections.

3.2 Problem Definition

Find a subset of edges $T \subseteq E$ such that:

1. (V, T) is a spanning tree.
2. $\deg_T(v) \leq b(v)$ for all $v \in V$.
3. The total cost $\sum_{e \in T} w(e)$ is minimized.

This problem is known as the **Degree-Constrained Minimum Spanning Tree (DC-MST)** problem.

4 Computational Complexity Analysis

In this section, the computational complexity of the DC-MST problem is rigorously studied, with the aim of theoretically justifying the algorithmic decisions adopted throughout the remainder of this work.

4.1 Theoretical Framework

This section provides a rigorous analysis of the computational complexity of the *Degree-Constrained Minimum Spanning Tree* (DC-MST) problem. The objective is to justify, from the standpoint of computational complexity theory, why it is not reasonable to expect efficient exact algorithms for its resolution in the general case, and to provide a theoretical foundation for the use of heuristics and approximation methods.

4.2 Membership of DC-MST in NP

A decision problem belongs to the class **P** if there exists a deterministic algorithm that solves it in polynomial time with respect to the input size. A problem belongs to the class **NP** if, given a candidate solution, it can be verified in polynomial time.

A problem is said to be **NP-hard** if every problem in **NP** can be reduced to it through a transformation computable in polynomial time. If, in addition, the problem belongs to **NP**, it is said to be **NP-complete**.

The classical Minimum Spanning Tree (MST) problem belongs to the class **P**. However, as shown below, the introduction of constraints on the maximum degree of vertices radically alters its computational complexity.

4.3 Canonical Problem Chosen: Hamiltonian Path

To prove the NP-hardness of the DC-MST problem, a polynomial-time reduction is employed from the canonical problem **Hamiltonian Path (HP)**.

Definition (Hamiltonian Path). Given an undirected graph $G = (V, E)$, determine whether there exists a simple path that visits all vertices exactly once.

This problem is known to be **NP-complete**.

The choice of Hamiltonian Path is natural for the following structural reasons:

- A path is a connected and acyclic graph, that is, a tree.
- In a path, internal vertices have degree exactly 2.
- The endpoints of the path have degree exactly 1.

These properties directly match the notion of a spanning tree with degree constraints, particularly when the degree bound is equal to 2.

4.4 Construction of the Reduction

Let $G = (V, E)$ be an arbitrary instance of the Hamiltonian Path problem. From this instance, a corresponding instance of the DC-MST problem is constructed as follows:

- Define $V' = V$.
- Define $E' = E$.
- For every edge $e \in E'$, assign a unit weight: $w(e) = 1$.
- Define the degree bound for all vertices as:

$$b(v) = 2 \quad \forall v \in V'.$$

The resulting instance consists of the weighted graph $G' = (V', E', w)$ together with the degree bound function b.

4.5 Correctness of the Reduction

We now prove that the constructed instance of DC-MST has a solution if and only if the original instance of Hamiltonian Path has a solution.

Forward implication $(\Rightarrow)$ Assume that a Hamiltonian path exists in the original graph G. By definition, such a path:

- Visits all vertices exactly once.
- Is connected and contains no cycles.
- Contains exactly $|V| - 1$ edges.

Moreover, in a Hamiltonian path each vertex has degree at most 2. Therefore, this path constitutes a valid spanning tree for the DC-MST instance, satisfies all imposed degree constraints, and its total cost is:

$$(|V| - 1) \cdot 1 = |V| - 1.$$

Consequently, a valid solution exists for the constructed DC-MST instance.

Backward implication $(\Leftarrow)$ Now assume that a valid solution exists for the constructed DC-MST instance. Such a solution is a spanning tree T that connects all vertices and satisfies:

$$\deg_T(v) \leq 2 \quad \forall v \in V'.$$

A tree in which all vertices have degree at most 2 can only have a path structure. Acyclicity prevents alternative configurations, and the degree constraint excludes any branching. Therefore, the tree T defines a simple path that visits all vertices exactly once.

Consequently, a Hamiltonian path exists in the original graph G.

4.6 Complexity of the Transformation

The described transformation does not add vertices or edges to the original graph. The assignment of weights and degree bounds is performed by traversing the sets V and E, and thus its time complexity is:

$$O(|V| + |E|).$$

Therefore, the reduction is computable in polynomial time.

4.7 Conclusion

Since Hamiltonian Path is an NP-complete problem and there exists a polynomial-time reduction $HP \leq_p DC\text{-}MST$, it follows that the DC-MST problem is **NP-hard**. This result provides the theoretical foundation for the approach adopted in the remainder of this work, which is based on the use of heuristics and approximation methods.

5 Algorithm Design

5.1 Exact Algorithm (Brute Force)

The exact algorithm enumerates all possible spanning trees, checks degree constraints, and selects the minimum-cost feasible solution. Although optimal, its exponential complexity limits its applicability to very small instances.

5.2 Greedy Heuristic

A greedy heuristic inspired by Kruskal's algorithm is employed. Edges are processed in ascending order of cost and added whenever they do not create cycles or violate degree constraints.

5.3 Local Search Improvement

The greedy solution is refined using a local search procedure based on edge exchanges. This approach improves solution quality while maintaining feasibility.

5.4 Metaheuristics for the DC-MST Problem

Due to the NP-hardness of the problem, metaheuristics are introduced to enhance global exploration of the solution space. Two approaches are considered: Simulated Annealing and Tabu Search.

5.4.1 Simulated Annealing

Simulated Annealing allows controlled acceptance of worse solutions based on a temperature parameter, enabling escape from local optima during early stages of the search.

Algorithm 1 Simulated Annealing for DC-MST

Initial solution T, graph G, degree bounds $b(v)$ Best solution T^* $T^* \leftarrow T$ temperature $> T_{min}$ Generate neighbor T' T' feasible $cost(T') < cost(T)$ $T \leftarrow T'$ Accept T' with probability $e^{-\Delta/T}$ $cost(T) < cost(T^*)$ $T^* \leftarrow T$ Decrease temperature T^*

5.4.2 Tabu Search

Tabu Search extends local search by incorporating memory structures that prevent cycling and promote exploration of new regions.

Algorithm 2 Tabu Search for DC-MST

Initial solution T, graph G, degree bounds $b(v)$ Best solution T^* $T^* \leftarrow T$ maximum iterations Generate feasible neighborhood Select best non-tabu neighbor Update tabu list $cost(T) < cost(T^*)$ $T^* \leftarrow T$ T^*

6 Implementation

All algorithms were implemented in Python using efficient data structures for cycle detection and degree tracking. A random instance generator was developed to ensure reproducibility.

7 Experimental Analysis

7.1 Methodology

Experiments were conducted on randomly generated instances with varying numbers of vertices, graph densities, and degree bounds. All algorithms were evaluated under the same conditions.

7.2 Metrics

- Total solution cost.
- Execution time.
- Approximation ratio (when optimal solution is available).

7.3 Results and Discussion

The results confirm the exponential growth of the exact algorithm and the scalability of heuristic and metaheuristic approaches. Simulated Annealing and Tabu Search consistently produced lower-cost solutions than classical heuristics at the expense of increased, but manageable, execution time.

8 Conclusions

This work demonstrates the gap between theoretical optimality and practical efficiency in NP-hard problems. By combining classical algorithm design with metaheuristics, it is possible to obtain high-quality solutions suitable for real-world network design scenarios.

9 Future Work

Future extensions include the use of advanced metaheuristics, approximation algorithms with theoretical guarantees, and hybrid approaches incorporating machine learning to guide heuristic decisions.