Project Report of COMP6651: Analyzing and Implementing Minimum-Cost Flow Algorithms on Randomized Source-Sink Networks

Shanmukha Venkata Naga Sai Tummala
Department of Computer Science
Concordia University, 40289721
shanmukha.tummala@outlook.com

Mohamed Mohamed
Department of Computer Science
Concordia University, 40266483
Mohamed.rasmy.fathy@gmail.com

Parsa Kamalipour
Department of Computer Science
Concordia University, 40310734
parsa.kamalipour@mail.concordia.ca

Naveen Rayapudi
Department of Computer Science
Concordia University, 40291526
rayapudinaveen777@gmail.com

Abstract

This project examines the minimum-cost flow problem in source-sink networks, an extension of the maximum-flow problem. Using adaptations of the Ford-Fulkerson method, we implement and evaluate four algorithms: Successive Shortest Path, Capacity Scaling, combined approach, and Additionally, we include the Primal-Dual Algorithm for comparison. Randomly generated directed Euclidean graphs with specific properties are used as test cases. For each graph, we identify the largest connected component and determine source-sink pairs algorithmically. Performance metrics, such as flow, cost, and path characteristics, are analyzed for eight predefined graph configurations and additional custom scenarios. This work aims to assess the efficiency and effectiveness of these algorithms under different conditions, providing insights into their computational performance and practical applications.

1 Introduction

1.1 Introduction to minimum-cost flow problem

The minimum-cost flow (MCF) problem is an important topic in network optimization, with applications in areas such as transportation, telecommunications, scheduling, and resource management. The goal of this problem is to find the cheapest way to transport a given amount of flow from supply nodes to demand nodes in a directed network. Each connection (arc) in the network has a capacity limit and a cost per unit of flow, and the objective is to minimize the total cost while meeting all demands [1].

Researchers have developed many algorithms to solve the MCF problem over the years. These methods include basic techniques like cycle-canceling and shortest-path algorithms, as well as advanced methods such as cost-scaling and network simplex algorithms. The performance of these algorithms can vary depending on the size, density, and structure of the network. For example, network simplex algorithms often work better on smaller networks, while cost-scaling algorithms are faster on large and sparse networks due to their better efficiency for such cases [2].

In this project, we study four algorithms to solve the MCF problem: Successive Shortest Path, Capacity Scaling, a hybrid method, and the Primal-Dual Algorithm. We test these algorithms on different types of networks, including both randomly generated and real-world examples. The aim is to understand how these methods perform under various conditions and provide practical insights into choosing the best algorithm for different types of problems.

1.2 Problem Description

The minimum-cost flow (MCF) problem is a fundamental challenge in network optimization. It involves determining the most cost-effective way to transport a specified amount of flow from supply nodes to demand nodes in a directed network. Each arc in the network has two key attributes:

- **Capacity** (u_{ij}): The maximum amount of flow that can pass through the arc.
- **Cost** (c_{ij}): The cost incurred per unit of flow on the arc.

Each node in the network is assigned a supply value (b_i), where:

- $b_i > 0$: The node is a supply node with a surplus of b_i.
- $b_i < 0$: The node is a demand node requiring $|b_i|$.
- $b_i = 0$: The node is a transshipment node with no supply or demand.

The objective is to find a flow assignment (x_{ij}) for all arcs (i, j) such that:

1. **Capacity Constraints:** The flow on each arc does not exceed its capacity:
$$0 \leq x_{ij} \leq u_{ij}, \quad \forall (i, j) \in A.$$
2. **Flow Conservation:** For every node, the total incoming flow plus supply equals the total outgoing flow:
$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i, \quad \forall i \in V.$$

The goal is to minimize the total cost of the flow:

$$TotalCost = \sum_{(i,j)\in A} c_{ij} \cdot x_{ij}.$$

The goal of the MCF problem is to minimize this total cost while satisfying all capacity and flow conservation constraints. [3]

This problem is a generalization of the maximum-flow problem, which focuses solely on maximizing flow without considering costs. As such, it has numerous practical applications, including transportation logistics, telecommunications network design, and resource allocation in supply chains. Solving this problem requires efficient algorithms that can handle both small-scale and large-scale network instances.

1.3 Motivation, Challenges, and Purpose of This Project

The minimum-cost flow (MCF) problem is important in network optimization, with applications in areas like transportation, telecommunications, and logistics. Solving this problem in real-world scenarios is difficult because networks are often large and complex. This project is motivated by the need to study and evaluate how different algorithms perform in terms of speed, scalability, and accuracy.

A major challenge is balancing computational speed with solution accuracy. Algorithms must handle large data and meet constraints on capacity, cost, and flow conservation. Their performance often depends on network features like size and density, making it essential to study when specific algorithms work best.

The purpose of this project is to implement and analyze four prominent algorithms for solving the MCF problem: the Successive Shortest Path, Capacity Scaling, Primal-Dual, and a hybrid approach. By testing these algorithms on randomly generated, this project aims to provide a comprehensive comparison of their efficiency, scalability, and practical applicability based on the Minimization results of each of the algorithms. Furthermore, the insights gained from this study will contribute to the broader understanding of network optimization methods and offer guidance for selecting appropriate algorithms for specific applications.

2 Related Work

2.1 Overview

2.2 Optimization-Based Methods

2.3 Modularity-Based Methods

2.4 Deep Learning Methods

3 Proposed Approach

3.1 GNN Architectures

3.1.1 GCNConv

3.1.2 GraphSAGEConv

3.2 Scalability Techniques

3.2.1 Full-Batch Training

3.2.2 Neighbor Sampling

3.2.3 Graph Partitioning

4 Results, Training, and Evaluation

4.1 Datasets

4.2 Training Approaches

4.3 Evaluation Metrics

4.4 Experimental Results

4.4.1 Synthetic Graphs

4.4.2 CORA Dataset

4.4.3 Reddit Dataset

4.4.4 Performance Summary Tables

5 Conclusion

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. Network flows: Theory, algorithms, and applications. 1993.

[2] P. T. Sokkalingam, Ravindra K. Ahuja, and James B. Orlin. New polynomial-time cycle-canceling algorithms for minimum-cost flows. *Networks*, 36:53–63, 2000.

[3] Péter Kovács. Minimum-cost flow algorithms: an experimental evaluation. *Optimization Methods and Software*, 30(1):94–127, 2015.