Algorithms for solving maximum flow problem

Members: Siqing Hou, Lu Yu, Zihao Wang

Introduction

Network flow problem is a kind of classical network optimization problem. We could find its applications in network packets routing, transportation scheduling, bridges destroy with minimum cost etc. Usually, a maximum flow problem needs to be solved in order to optimize practical cost for certain task. Formally, maximum flow problem [3] can be formulated as finding a flow from source node s to target node t with a given directed graph G = (V, E), where each edge e is associated with its capacity c(e) > 0. There are two constraints: (i) flow on each edge doesn't exceed c(e). (ii) for every node v non-s and t, incoming flow is equal to outgoing flow. According to [3], many pioneering research have provided efficient solutions to maximum flow problem. In this project, our main job is to fully understand this problem, and conduct some experiments to explore different aspects of two popular algorithms.

Algorithms

Edmonds–Karp algorithm [1]: an implementation of the Ford–Fulkerson method for computing the maximum flow in a flow network in $O(VE^2)$ time. It's a specialization of Ford–Fulkerson, finding augmenting paths with breadth-first search.

Dinic's Blocking flow algorithm[2]**:** The algorithm runs in $O(V^2E)$ time and is similar to the Edmonds–Karp algorithm, which runs $O(VE^2)$ time, in that it uses shortest augmenting paths. The introduction of the concepts of the level graph and blocking flow enable Dinic's algorithm to achieve its performance.

References:

[1] Jack Edmonds, Richard M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*19.2 (1972): 248-264.
[2] E. A. Dinic, Algorithm for solution of a problem of maximum flow in a network with power estimation, Soviet Math. Doll. 11 (5), pp. 1277-1280,(1970).
[3] https://en.wikipedia.org/wiki/Maximum_flow_problem