# Report
# Combinatorial Optimization

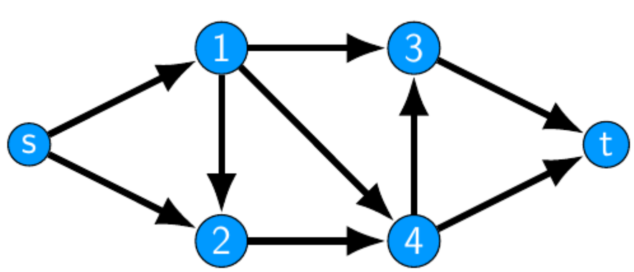
# A. Maximum Network Flow

## 1. Introduction

- The maximum flow problem was first formulated in 1954 by T. E. Harris and F. S. Ross as a simplified model of Soviet railway traffic flow. In 1955, Lester R. Ford, Jr. and Delbert R. Fulkerson created the first known algorithm, the Ford–Fulkerson algorithm.
- Over the years, various improved solutions to the maximum flow problem were discovered, notably the shortest augmenting path algorithm of Edmonds and Karp and independently Dinitz; the blocking flow algorithm of Dinitz; the push-relabel algorithm of Goldberg and Tarjan; and the binary blocking flow algorithm of Goldberg and Rao. The algorithms of Sherman and Kelner, Lee, Orecchia and Sidford, respectively, find an approximately optimal maximum flow but only work in undirected graphs.

## 2. Related Work

- Input: a connected digraph G=(V,E) and edge capacity.
- Task: find a feasible s-t flow of maximum value




Define the Manimux Cut and Minimum Cut

# B. Shortest Path

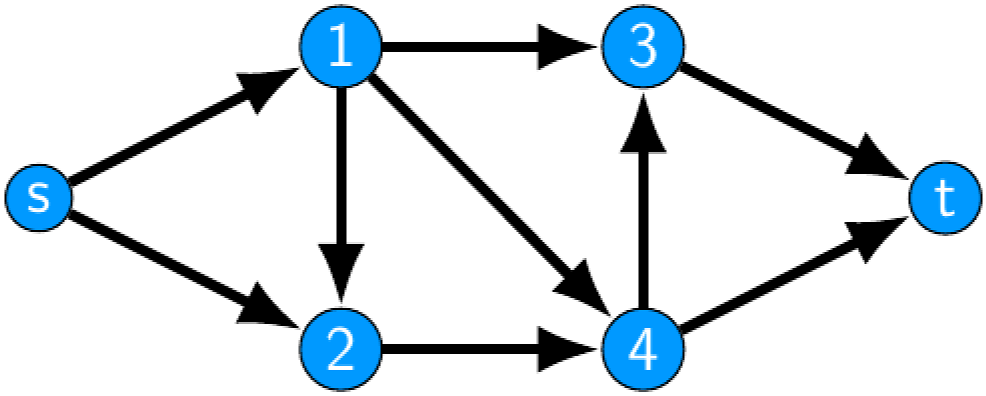
## 1. Introduction

Generally, in order to represent the shortest path problem we use graphs. A graph is a mathematical abstract object, which contains sets of vertices and edges. Edges connect pairs of vertices. Along the edges of a graph it is possible to walk by moving from one vertex to other vertices. Depending on whether or not one can walk along the edges by both sides or by only one side determines if the graph is a directed graph or an undirected graph.

- In addition, lengths of edges are often called weights, and the weights are normally used for calculating the shortest path from one point to another point. There exist different types of algorithms that solve the shortest path problem: Dijkstra's Algorithm,

## 2. Related Work

- Input: a connected (di-)graph G=(V,E) an edge valuation c:E->R, distinct nodes s,t  V s, t ϵ V
- Task: find a shortest path connecting s and t in G with respect to c



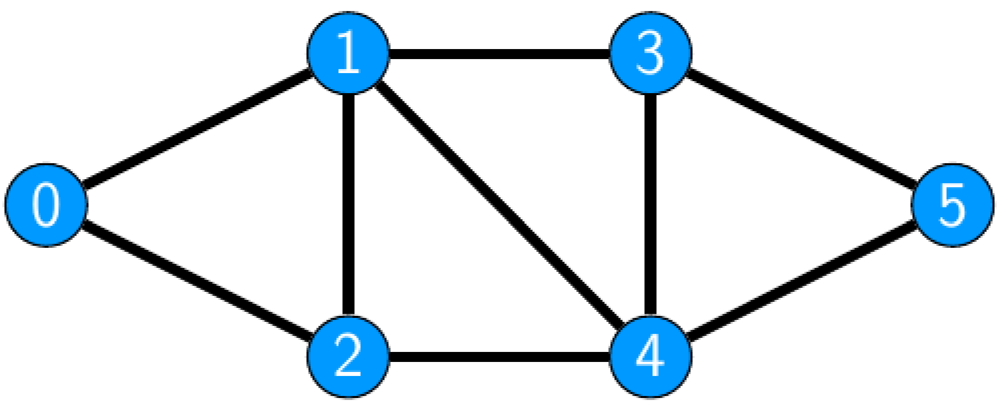

Example: Finding the Shortest-Path

# C. Minimum Spanning Tree

## 1. Introduction

## 2. Related Work

- Input: a connected graph G=(V, E) and an edge valuation c:E->R.
- Task: find a spanning tree T⊂ E of minimal total weight c(T)




Example: Minimum Spanning Tree

# References:

https://en.wikipedia.org/wiki/Maximum_flow_problem

https://en.wikipedia.org/wiki/Shortest_path_problem
https://en.wikipedia.org/wiki/Minimum_spanning_tree