**Project:**

**Combinatorial Optimization**

**A. Basic Approaches**

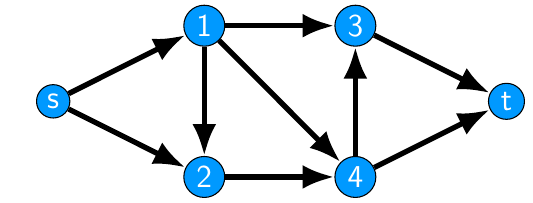
Description: You need to define three basic algorithms based on solving the combinatorial optimization, included maximum network flow, shortest path, and minimum spanning tree

**A.1 Maximum Network Flow**

+ Problem (Maximum Network Flow)

input: a connected digraph , and edge capacity.

Task: find a feasible s-t flow of maximum value

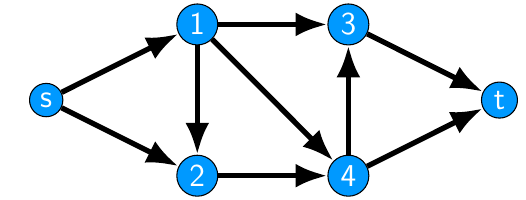


Define the Manimux Cut and Minimum Cut

**A.2 Shortest Path**

Input: a connected (di-)graph an edge valuation , distinct nodes s,t V

Task: find a shortest path connecting s and t in G with respect to

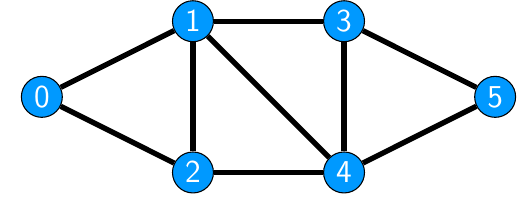


Example: Finding the Shortest-Path

**A.3 Minimum Spanning Tree**

Input: a connected graph and an edge valuation .

Task: find a spanning tree of minimal total weight



Example: Minimum Spanning Tree

**Implementation:**

+ Implement the three basic algorithms that can be adaptive to any input data change.

+ read input data from the nodes and edges, and predict the minimum and maximum cut and also predicting the shortest path

+ design the interface for input from user by representing the feature like: Add, Insert, Delete, Dropout, Change parameters, ...

+ visualize the networks

Requirements:

- Any copy from Internet will be got zero points

- Any images or referred must be reference cited in your reports

**B. Combinatorial Optimization with Neural Networks (Advance Topic for excellent students)**

* Define the forward and backward of network(e.g., expected 1 input, 2 hidden layer, 1 output) [ 3, 4, 5, 6, 7, 8, 9, 10]

Implementation: you are referred to implement in Python to reduce the complexity.

Reference:

[1] Minimum spanning tree: <https://en.wikipedia.org/wiki/Minimum_spanning_tree>

[2] Maximum flow problem: <https://en.wikipedia.org/wiki/Maximum_flow_problem>

[3] Yujia Li, Daniel Tarlow, Marc Brockschmidt, Richard Zemel, GATED GRAPH SEQUENCE NEURAL NETWORKS, arXiv:1511.05493

[4] Michaël Defferrard, Xavier Bresson, Pierre Vandergheynst, Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, arXiv:1606.09375.

[5] Kipf, Thomas N and Welling, Max, Semi-Supervised Classification with Graph Convolutional Networks, arXiv preprint arXiv:1609.02907, 2016.

[7] [http://mat.uab.es/~alseda/MasterpOst/PotvinSmith\_NeuralNetworks-Corrected.pdf](http://mat.uab.es/~alseda/MasterOpt/PotvinSmith_NeuralNetworks-Corrected.pdf)

[8] Bello, I., Pham, H., Le, Q. V., Norouzi, M., & Bengio, S. (2016). Neural combinatorial optimization with reinforcement learning. arXiv preprint arXiv:1611.09940.

[5] Azalia Mirhoseini, Hieu Pham, Quoc Le, Mohammad Norouzi, Samy Bengio, Benoit Steiner, Yuefeng Zhou, Naveen Kumar, Rasmus Larsen, and Jeff Dean, Device placement optimization with reinforcement learning, 2017.

[9] Dai, Hanjun and Khalil, Elias B and Zhang, Yuyu and Dilkina, Bistra and Song, Le, "Learning Combinatorial Optimization Algorithms over Graphs", arXiv preprint arXiv:1704.01665, 2017. [https://papers.nips.cc/paper/7214-learning-combinatorial-optimization-algorithms-over-graphs.pdf]