The Source Code of the Two Heuristic Algorithms:

Maximum Adjacency Search Algorithm

The files in the folder "./SourceCode/" are the C++ source code of the maximum adjacency search algorithm, which is shown in Algorithm 4 in the paper.

This program requires that the subgraph induced by the set of input seed nodes should be connected. If not, please use the Mehlhorn's algorithm [26] to compute the Steiner tree given the query nodes.

1. Arguments in the Input Command

When you compile the project and get a ".exe" file, you may use the following input arguments.

Table 1. The arguments

-n	the input file name: nodes of the network
-w	the input file name: query biased node weights
-e	the input file name: edges of the network
-s	the input file name: seed nodes
-o	the output folder
-k	the parameter K in the algorithm

The following is one example input command.

-n ./ToyGraph/01Nodes.txt -e ./ToyGraph/02Edges.txt -w ./ToyGraph/03NodeWeights.txt -s ./ToyGraph/04Seeds.txt -o ./ToyGraph/Output/ -k 40

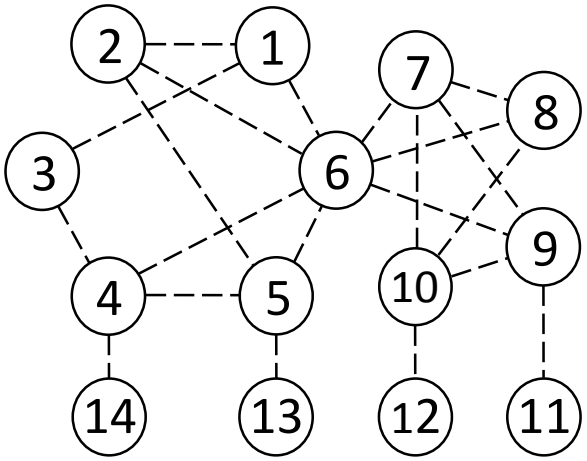
This input command means that the input dual networks' node file is "./ToyGraph/01Nodes.txt", the input network edge file is "./ToyGraph/02Edges.txt", the input node weight file is "./ToyGraph/03NodeWeights.txt", the input seed node file is "./ToyGraph/04Seeds.txt", the output folder is "./ToyGraph/Output/", and the parameter k is set to 40.

2. Formats of the Input Files

In the node file, each row contains the name of one node.

In the edge file, each row contains one edge. Each row contains three columns: index of the first node, index of the second node, and the edge weight. The index of node represents the row number of the node in the node file minus one. The three columns are separated by the space symbol.

One example of dual networks is shown in the following figures.



Conceptual network

The node and edge files are shown in the following table.

node file	node weight	edge file
1 2 3 4 5 6 7 8 9 10 11 12 13 14	1 1 1 1 1 1 1 1 1 1 1 1 1 1	0 1 1.0 0 2 1.0 0 5 1.0 1 4 1.0 1 5 1.0 2 3 1.0 3 4 1.0 3 5 1.0 3 13 1.0 4 5 1.0 4 12 1.0 5 6 1.0 5 7 1.0 5 8 1.0 6 7 1.0 6 8 1.0 6 9 1.0 7 9 1.0 8 9 1.0 8 10 1.0 9 11 1.0

3. Format of the Output File

The output file will be in the folder designated by the input argument "-o". The files have the same formats. In the above example, the subgraph of "1,2,3,4,5,6,7,8,9,10" is the densest subgraph. The output files for the densest subgraph are shown in the following table.

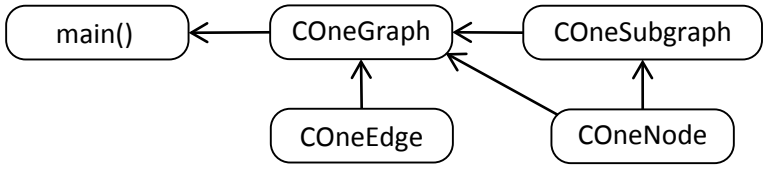
node file	edge file
1 2 3 4 5 6 7 8 9 10	0 1 1 0 2 1 0 5 1 1 4 1 1 5 1 2 3 1 3 4 1 3 5 1 4 5 1 5 6 1 5 7 1 5 8 1 6 7 1 6 8 1 6 9 1 7 9 1 8 9 1

The summary of the densest connected subgraph:

Number of Nodes : 10
Number of Edges : 17
Density of the Subgraph : 1.7
Maximum Assigned Edge Weight : 3
Approximation Ratio : 1.76471

4. Design of the Classes

The C++ source code of the algorithm is the Windows version. The "MainEntrance.cpp" file contains the "main()" function. The class design is shown in the following figure. The arrow from A to B represents that the class A is used by B.



This project contains 4 classes: COneGraph, COneNode, COneEdge, and COneSubgraph. COneGraph is the main class. It contains the nodes and edges of the graph. It will also read the

input files and call the DCS_GND algorithm. COneNode contains the information of one node, such as the name of the node, the adjacent nodes of this node. COneEdge contains the information of one edge, and it is a quite simple class. COneSubgraph contains the subgraphs induced by the remaining nodes in the maximum adjacency search process. The kernel of the algorithm is implemented in this class, i.e., in the “OneSubgraph.cpp” file.