The Source Code of the DCS_GND Algorithm

The files in the folder "./SourceCode/" are the C++ source code of the DCS_GND algorithm, which is Algorithm 2 of the paper. The algorithm can also handle the seed nodes and size constraint.

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
-p	the input file name: edges of the physical network
-c	the input file name: edges of the conceptual network
-s	the input file name: seed nodes
-o	the output folder
-g	the parameter γ in Algorithm 2 of the paper
-k	the size constraint in the DCS_k problem

The following is one example input command.

-n ./ToyDualGraphs/01Nodes.txt -p ./ToyDualGraphs/02EdgesP.txt -c ./ToyDualGraphs/03EdgesC.txt -s ./ToyDualGraphs/04Seeds.txt -o ./ToyDualGraphs/Output/ -g 1.5 -k 40

This input command means that the input dual networks' node file is "./ToyDualGraphs/01Nodes.txt", the input physical network edge file is "./ToyDualGraphs/02EdgesP.txt", the input conceptual network edge file is "./ToyDualGraphs/03EdgesC.txt", the input seed node file is "./ToyDualGraphs/04Seeds.txt", the output folder is "./ToyDualGraphs/Output/", the parameter γ is set to 1.5, and the parameter k is set to 40. The resulting DCS will contain 40 nodes including the seed nodes in the "04Seeds.txt" file. If there is no seed node, we can set the "04Seeds.txt" file as an empty file. If there is no size constraint, we can simply ignore the input parameter "-k".

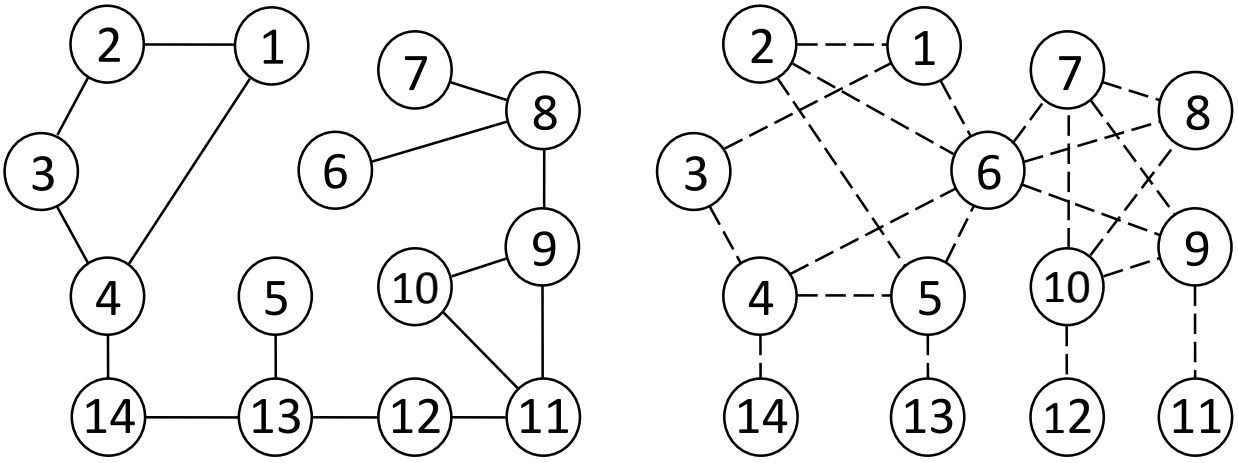
2. Formats of the Input Files

In the node file, each row contains the name of one node. One example is shown as follows.

In the physical network edge file, each row contains one edge. Each row contains two columns: index of the first node and index of the second node. The index of node represents the row number of the node in the node file minus one. 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.

In the conceptual network 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.



(a) Physical network (b) Conceptual network

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

node file	physical network edge file	conceptual network edge file
1 2 3 4 5 6 7 8 9 10 11 12 13 14	0 1 0 3 1 2 2 3 3 13 4 12 5 7 6 7 7 8 8 9 8 10 9 10 10 11 11 12 12 13	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 "6,7,8,9,10" is the densest connected subgraph. The output files for the densest connected subgraphs are shown in the following table.

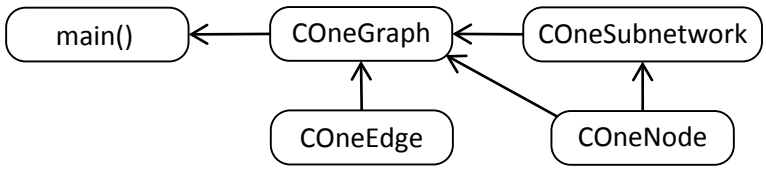
node file	physical network edge file	conceptual network edge file
6 7 8 9 10	0 2 1 1 2 1 2 3 1 3 4 1	0 1 1 0 2 1 0 3 1 1 2 1 1 3 1 1 4 1 2 4 1 3 4 1

The summary of the densest connected subgraph:

Number of Nodes : 5
Number of Edges in the Physical Network : 4
Number of Edges in the Conceptual Network : 8
Density of the Subgraph in the Conceptual Network : 1.6
Maximum Assigned Edge Weight : 3
Approximation Ratio : 1.875

4. Design of the Classes

The C++ source code of the DCS_GND 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 COneSubnetwork. 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. COneSubnetwork contains the subgraphs induced by the remaining nodes in the greedy node deletion process. The kernel of the algorithm is implemented in this class, i.e., in the "OneSubnetwork.cpp" file.