

How to Use the Source Code?

The files in the folder “./SourceCode/” are the C++ source code of the FLoS_PHP algorithm.

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 graph
-e	the input file name: edges of the graph
-s	the input file name: seed nodes
-o	the output folder
-c	the decay factor in PHP
-k	the number k for the top- k query
-u	the value used in the termination criterion

The following is one example input command.

```
-n C:\01Nodes.txt -e C:\02Edges.txt -s C:\03Seeds.txt -o C:\ -c 0.5 -k 10 -u 0.001
```

This input command means that the input graph’s node file is “C:\01Nodes.txt”, the input graph’s edge file is “C:\02Edges.txt”, the input seed node file is “C:\03Seeds.txt”, the decay factor is 0.5, the number k is 10, and the termination criterion value is 0.001.

When the k th and $(k + 1)$ th nodes have close exact proximity values, the algorithm will visit many nodes to distinguish them. The value in “-u” is used to stop the program in this case, i.e., if the k th largest lower bound plus the value in “-u” is greater than the maximum upper bound of the remaining nodes, the algorithm will terminate.

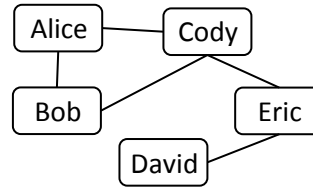
2. Formats of the Input Files

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

Alice
Bob
Cody
David
Eric

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 is shown in the following table (on the left).

0	1	0.3
0	2	0.4
1	2	0.2
2	4	0.7
3	4	0.4



The first row “0 1 0.3” represents the edge with weight 0.3 between two nodes “Alice” (with index 0) and “Bob” (with index 1). Based on the node and edge files, the graph is shown on the above figure (on the right).

In the seed node file, each row contains the name of one query node. One example is shown as follows. The algorithm will pick each query node (in each row) and run the FLoS_PHP algorithm to query its top- k nodes.

Alice
David

3. Format of the Output File

The output file will be in the folder designated by the input argument “-o”. Each row contains the names of the top- k nodes with regard to the query node. The names are separated by the space symbol. One example is shown as follows. In this example, we use the above graph, the above query, and $k = 2$.

Bob Cody
Eric Cody

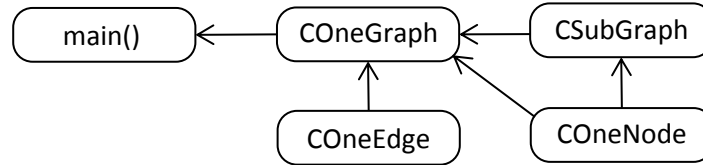
The first row represents that “Bob” and “Cody” are the top-2 nodes with regard to the query node “Alice”. The second row represents that “Eric” and “Cody” are the top-2 nodes with regard to the query node “David”.

The toy graph in the Figure 1(a) in the paper is in the “./ToyGraph/” folder. You may use the following input command:

```
-n ./01Nodes.txt -e ./02Edges.txt -s ./03Seeds.txt -o ./ -c 0.8 -k 2 -u 0.001
```

4. Design of the Classes

The C++ source code of the FLoS_PHP 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 CSubGraph. COneGraph is the main class. It contains the nodes and edges of the graph. It will also read the input files and call the FLoS_PHP algorithm. COneNode contains the information of one node, such as the name of the node, the adjacent nodes of this node, and the corresponding transition probabilities. COneEdge contains the information of one edge, and it is a quite simple class. CSubGraph contains the subgraph induced by the visited nodes in the local search process. The kernel of the algorithm is implemented in this class, i.e., in the “SubGraph.cpp” file.