MPRI Graph Mining: Find Quasi Cliques

Mohammad Mahzoun
Mohammad.Mahzoun@gmail.com

February 15, 2020

1 Introduction

Our aim in this project is to find $\alpha - quasi - cliques$ with at least s nodes using a given heuristic approach. In this document, the running time of the implementation for different graphs is reported.

1.1 Hardware

The system we are using to run the code is running a Ubuntu 18.04 operating system with 16GB of memory and Intel COREI7 8th Gen CPU.

1.2 Implementation

The code is publicly available on Github.

2 Results

In the following, we use different graphs as the input to algorithm. The values $k \in \{2, 3, 4\}$ and $s = 10$ are used for the required parameters. In each section, the results for each graph is described.

2.1 Pretty Good Privacy

This is the interaction network of users of the Pretty Good Privacy (PGP) algorithm. The network contains only the giant connected component of the network. The graph is undirected with 10680 nodes and 24316 edges.

k	$result$	$time$	$size$
2	1	$95s$	24
3	1	$98s$	24
4	1	$100s$	24

By increasing the value of k, the running time increase because the graph is sparse and calculating the $k - degree$ is slower by the increasing of k. The result is same because there is a complete sub-graph of order 24.

2.2 Route views

This is the undirected network of autonomous systems of the Internet connected with each other. Nodes are autonomous systems (AS), and edges denote communitation. The network contains loops. The graph is undirected with 6474 nodes and 13895 edges.

k	$result$	$time$	$size$
2	1	$25s$	10
3	1	$110s$	10
4	1	$250s$	10

By increasing the value of k, the running time increase because the graph is sparse and calculating the $k - degree$ is slower by the increasing of k. The result is same because there is a complete sub-graph of order 24.

2.3 US power grid

This undirected network contains information about the power grid of the Western States of the United States of America. An edge represents a power supply line. A node is either a generator, a transformator or a substation. The network contains loops. The graph is undirected with 4941 nodes and 6594 edges.

k	$result$	$time$	$size$
2	0.6	$10s$	10
3	0.57	$20s$	10
4	0.55	$28s$	10

By increasing the value of k, the running time increase because the graph is sparse and calculating the $k - degree$ is slower by the increasing of k. The result is different because the order of nodes in deleting has been changed.

3 Improvement

The bottleneck of the algorithm is listing all $k-cliques$. Minor implementation tunings will result in better performance. However, for small values of k, we may do better. For $k == 2$, we may just use the degree of each vertex which is faster.