Homework Assignment 5: Network Flows

5273

Introduction

For this work the library NetworkX of Python have been use to generate an undirected powerlaw cluster graph of 8 nodes [4]. This graph it is used to define the instances, characterized by the graph and different pairs of source and targets. Edge weights have been generated following a normal distribution with mean of 3 units and a standard deviation of 1.5. This data it is used to generate a system where a maximum flow problem needs to be solved.

This work it is run on an Intel Celeron CPU @ 1.10 GHz with 4 GB RAM laptop.

The powerlaw cluster graph has been chosen because it perfectly can modelate a logistic supply chain network where every node represents each logistic point and the edges the possible connections between every point[3]. The objective is to maximize the flow of value in the network[1].

In order to draw this situation a Fruchterman-Reingold layout is chosen, which seems to be the good way to understand the interactions among nodes because of the attempts to minimize the number of overlapping nodes and edges [5]. Also this layout allows to to distribute vertices evenly, make edge lengths uniform, and reflect symmetry [2].

The nodes are represented by different colors. The skyblue color represents the source nodes whereas the orange color represents the targets nodes. The rest of the nodes are represented in red. Another visual modification made is that sources and target nodes are represented with a bigger size than the other nodes.

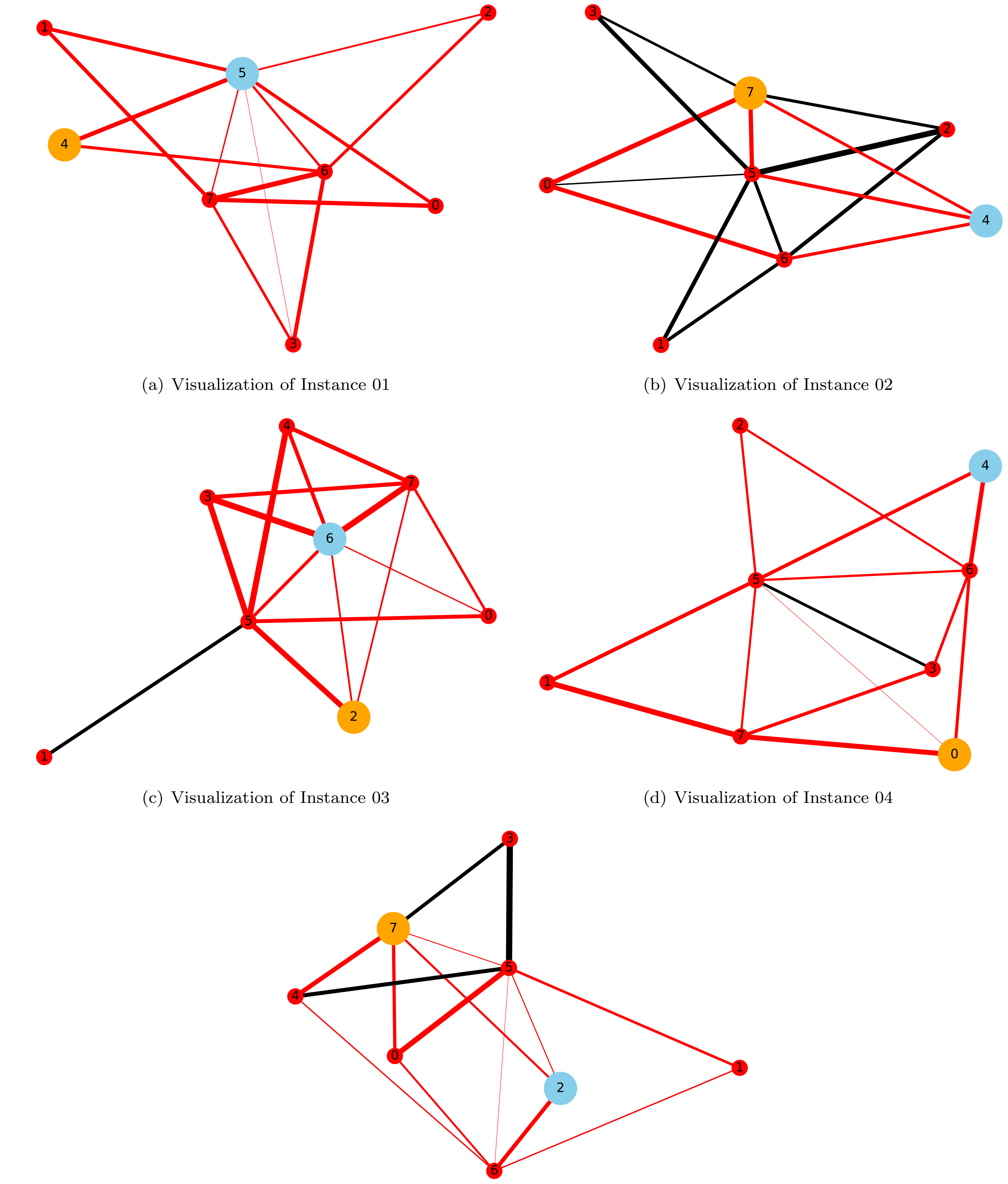
In the case of edges they are represented in black color. The width of the edge indicates its capacity, for example, wider edges are the ones with greater capacity. When the maximum flow algorithm is executed those edges where the flow passes are colored in red in order to indicate the maximum flow produced by the algorithm.

```python
for k in range(5):
    #Maximum flow algorithm:
    flow_value,flow_dict = nx.maximum_flow(G, st_list[k][0], st_list[k][1], capacity='weight')
    #print(flow_dict)

    #Change edge colors
    edge_colors = ['black' if flow_dict[i][j] == 0 and flow_dict[j][i] == 0 else 'red' for i,
    j in G.edges()]

    #Change node properties:
    color_map = []
    node_sizes = []
    for node in G:
        if node == st_list[k][0]:
            color_map.append('skyblue')
            node_sizes.append(900)

        elif node == st_list[k][1]:
            color_map.append('orange')
            node_sizes.append(900)
        else:
            color_map.append('red')
            node_sizes.append(200)

    #Layout:
        pos = nx.fruchterman_reingold_layout(G, k=0.2, iterations=40)

    nx.draw(G, node_color=color_map, edge_color=edge_colors,
            node_size=node_sizes, width=weights, pos=pos,
            with_labels=True)
    plt.show()
```

graphs_t5.py

Structural Characterization of Instances

Instances are characterized by a Graph, in this case the powerlaw cluster graph, by a source node and by a target node. For this work 5 instances are generated. The source and target nodes are selected at random.



(a) Visualization of Instance 01

(b) Visualization of Instance 02

(c) Visualization of Instance 03

(d) Visualization of Instance 04

(e) Visualization of Instance 05

Figure 1: Instances

This way the instances are:

- Instance 01: G, 5, 4.
- Instance 02: G, 4, 7.
- Instance 03: G, 6, 2.
- Instance 04: G, 4, 0.
- Instance 05: G, 2, 7.

For the visualized graphs the following 6 structural characteristics have been calculated:

- Degree distribution
- Clustering coefficient
- Closeness centrality
- Load centrality
- Eccentricity
- Page Rank

In order to analyze the influence of this characteristics in the execution time as well as in the optimum value boxplots are constructed.

Also an analysis of best sources and targets it is carried out based on the values of the maximum flow algorithm. Three nodes are given to conclude those best values.

For this analysis, within each instance a source node is fixed and the other nodes are variable, then the maximum flow algorithm is executed and the best three values will be considered as the better ones for that instance. The same procedure is followed for target nodes.

As a result the best target nodes for Instance 01 are the nodes 6, 7 and 1, whereas the best source nodes are 0, 1 and 2 if we want to maximize the flow of the network.

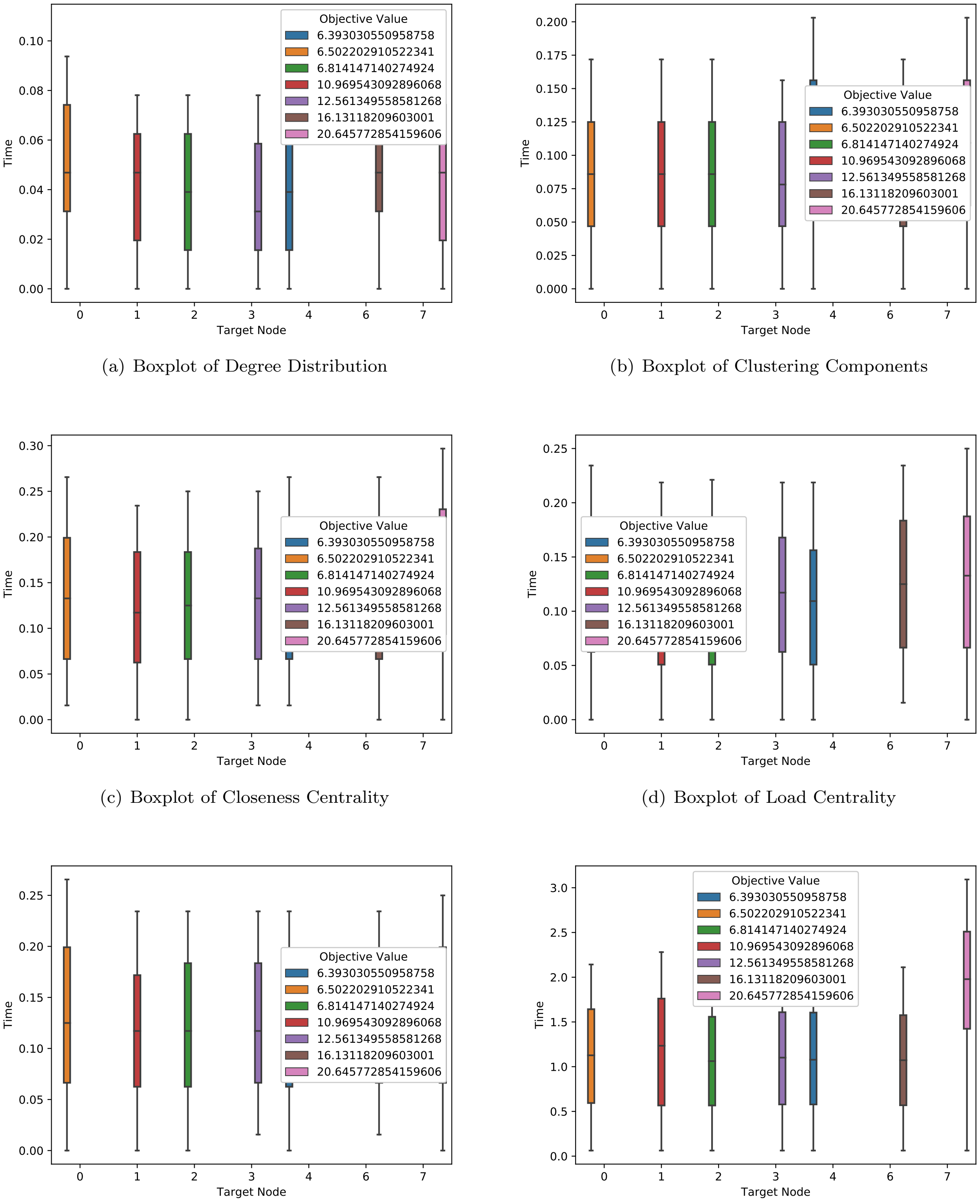
For Instance 02 the best targets are 0, 1 and 3, whereas nodes 6, 5 and 0 are the best sources

In case of Instance 03 nodes 5, 7 and 3 are the best targets and nodes 3, 4 and 5 are the best sources.

For Instance 04 the results are 3, 5 and 7 as best targets and 2, 3 and 4 as the best sources.

Last, for Instance 05 the best targets are 0, 1 and 3 whereas the best sources are 5, 6 and 0.

Moreover, it is important to investigate the effect of the structural characteristics of vertices in the computation time. In the case of Instance 01 Computation Time seems to be quite similar but with a little different in case of node 7 which have the greater objective value and the longest time. These results are shown in the figures bellow:



(a) Boxplot of Degree Distribution

(b) Boxplot of Clustering Components

(c) Boxplot of Closeness Centrality

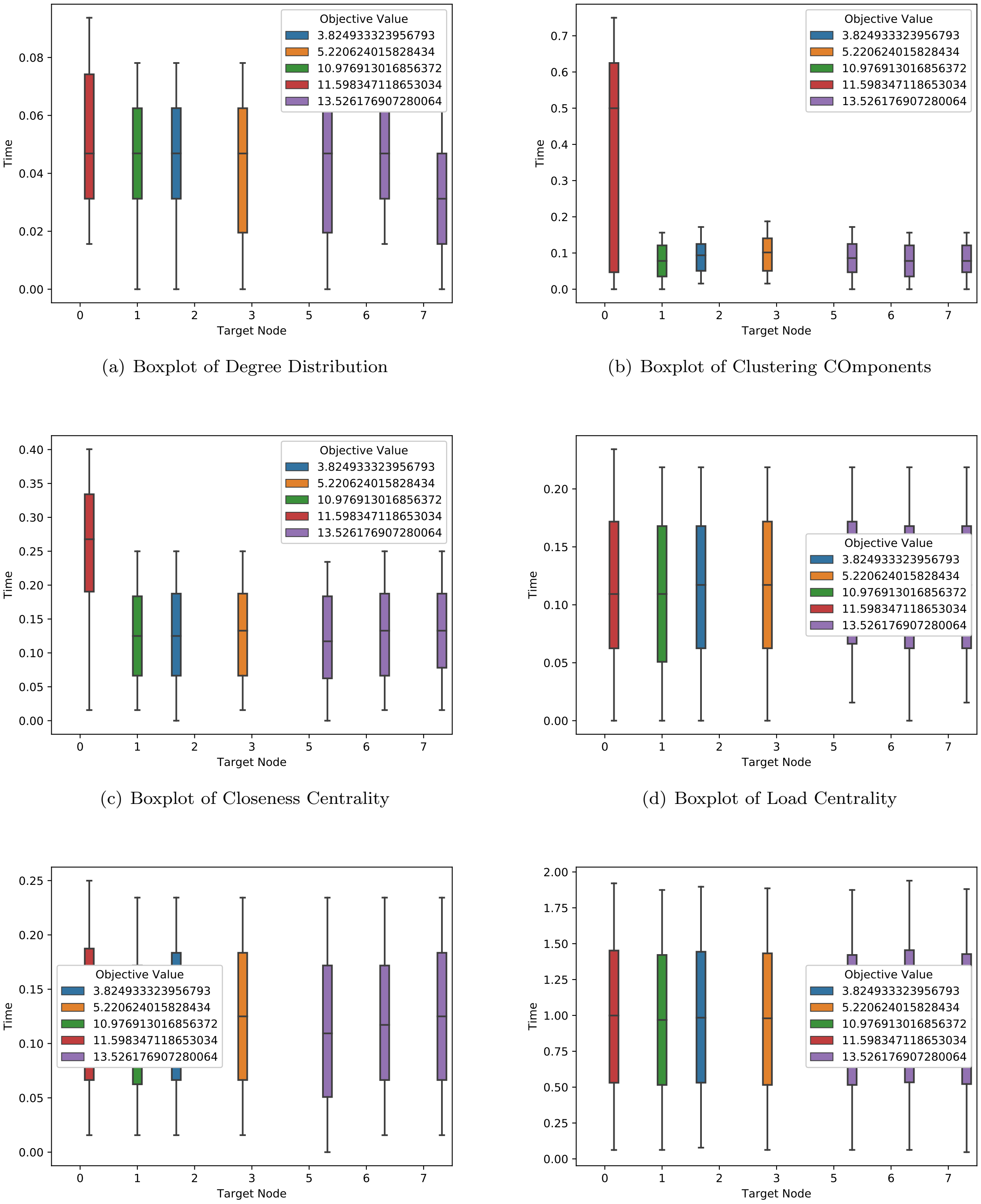
(d) Boxplot of Load Centrality

(e) Boxplot of Eccentricity

(f) Boxplot of Page Rank

Figure 2: Boxplots of Instance 01

For Instance 02 we have something similar like Instance 01, where time is different in case of some node 0, being the longest in the characteristics Degree Distribution, Clustering Componets and Closeness Centrality, the rest is practically the same. Also this node has the second highest flow value. Here the maximum flow values are lower than Instance 01.



(a) Boxplot of Degree Distribution

(b) Boxplot of Clustering COmponents

(c) Boxplot of Closeness Centrality

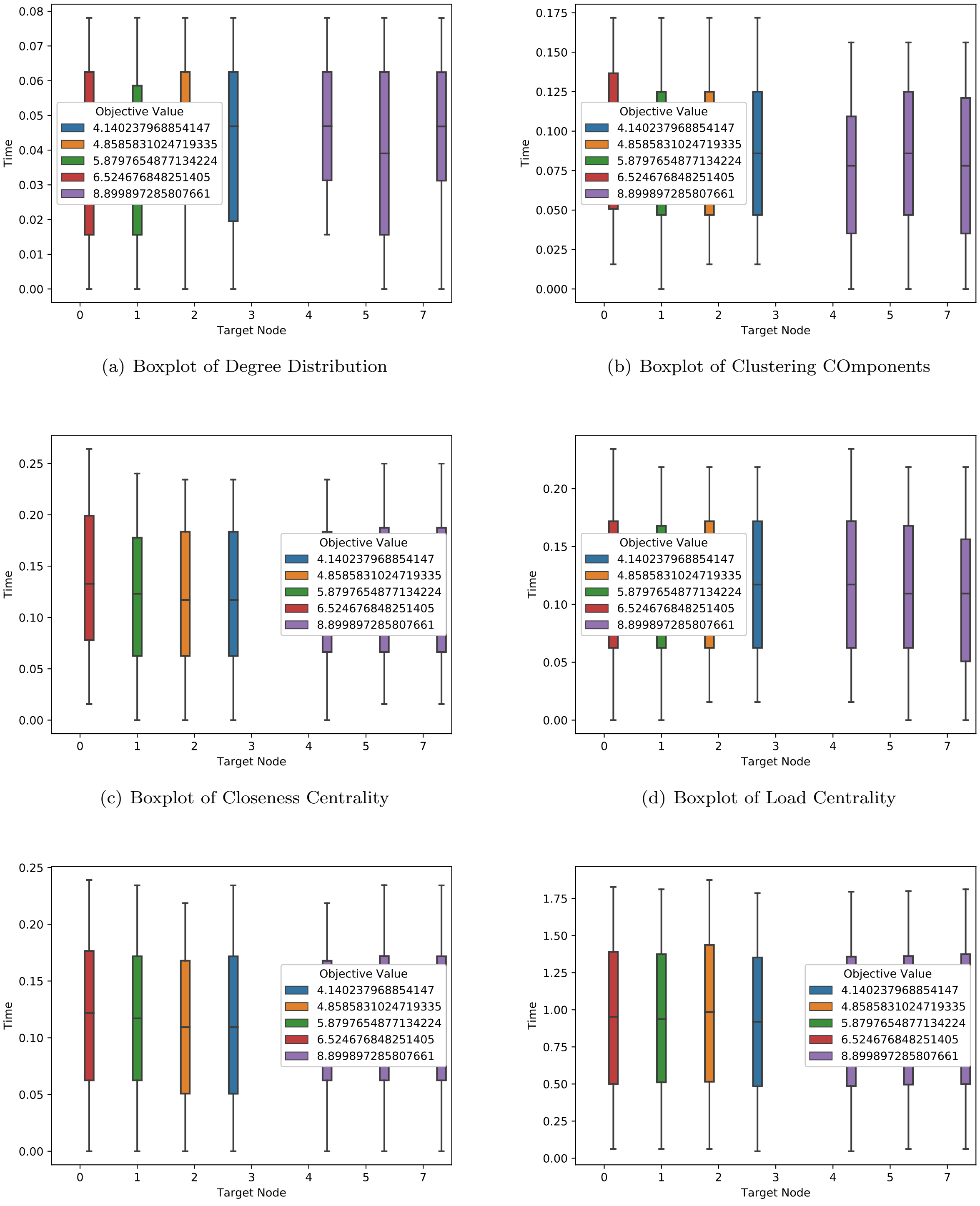
(d) Boxplot of Load Centrality

(e) Boxplot of Eccentricity

(f) Boxplot of Page Rank

Figure 3: Boxplots of Instance 02

For Instance 03 the time variable is almost the same in all cases, it could be due to the closest values of maximum flow, and also they are not high in comparison with the other instances.



(a) Boxplot of Degree Distribution

(b) Boxplot of Clustering COmponents

(c) Boxplot of Closeness Centrality

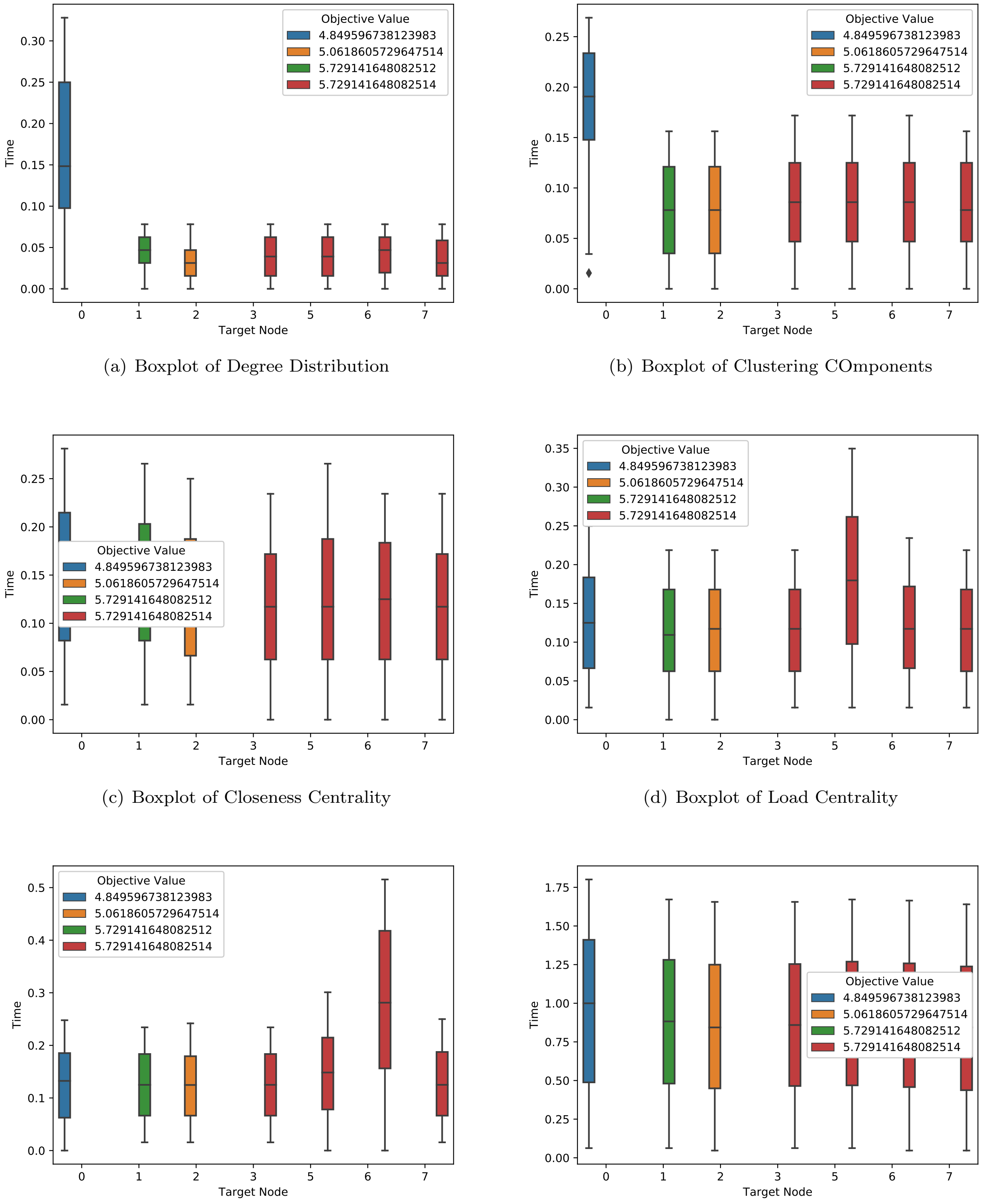
(d) Boxplot of Load Centrality

(e) Boxplot of Eccentricity

(f) Boxplot of Page Rank

Figure 4: Boxplots of Instance 03

In the case of Instance 04 it can be found the lowest values of flow for the instances tested. Here in some cases node 0 has some peacks in execution time in the case of Degree Distribution and Clustering Components. Also node 5 presents a higher time for Load Centrality and Eccentricity.



(a) Boxplot of Degree Distribution

(b) Boxplot of Clustering COmponents

(c) Boxplot of Closeness Centrality

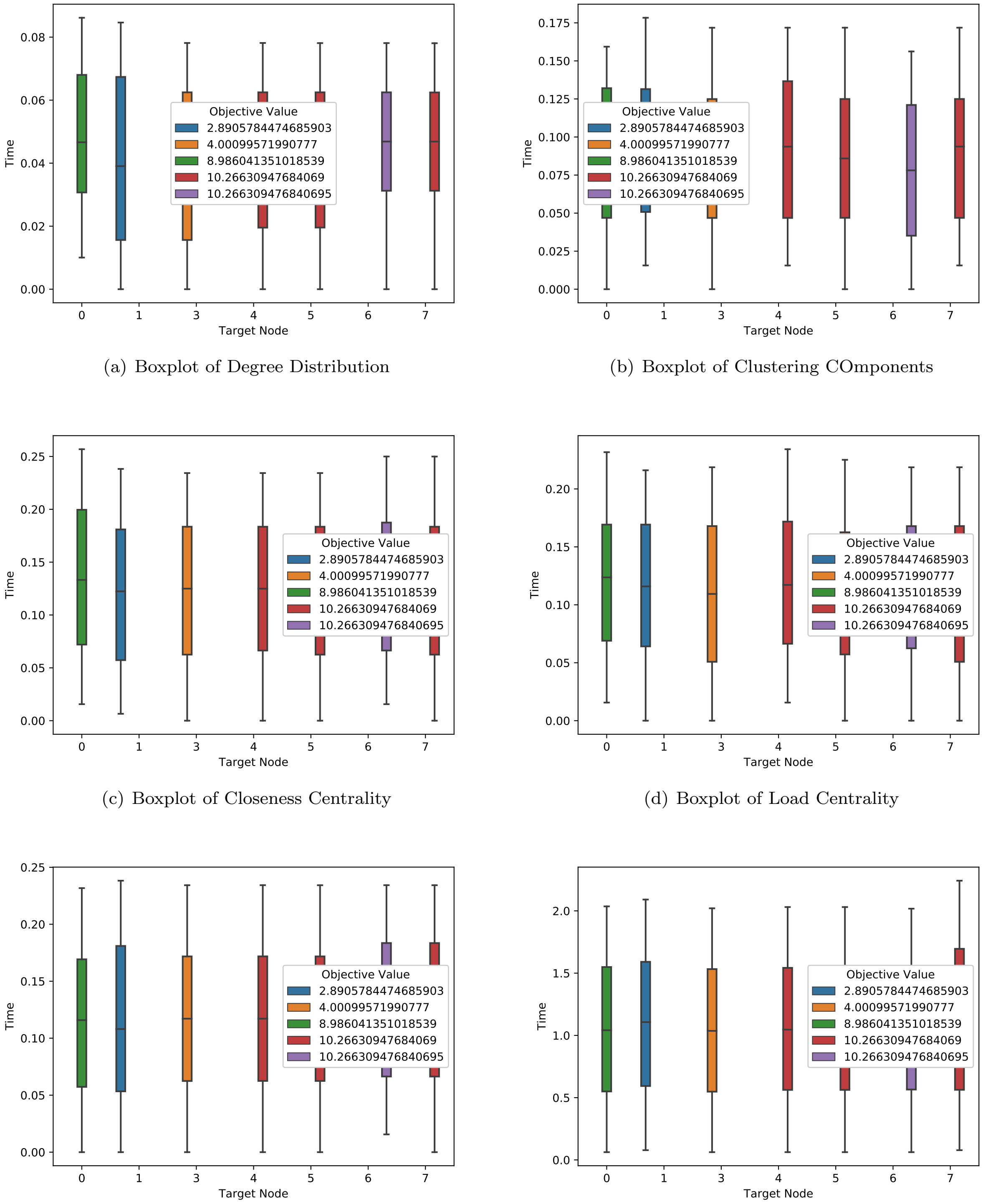
(d) Boxplot of Load Centrality

(e) Boxplot of Eccentricity

(f) Boxplot of Page Rank

Figure 5: Boxplots of Instance 04

For Instance 05 all data seems to be similar without any node standing out from others as the figure shows below.



(a) Boxplot of Degree Distribution

(b) Boxplot of Clustering COmponents

(c) Boxplot of Closeness Centrality

(d) Boxplot of Load Centrality

(e) Boxplot of Eccentricity

(f) Boxplot of Page Rank

Figure 6: Boxplots of Instance 05

As a conclution it can be said that the maximum flow value obtained is 20.65 units with Instance 01 being node 7 the target and node 5 the source. On the other hand Instance 03 is the one with lowest objectives values with nodes 2 and six Indicating they are not good target and source nodes for maximizing flow.

Page Rank and Closeness centrality are the slowest characteristics in terms of execution time, whereas the fastest characteristics are Degree distribution and Clustering Components but we can not say there is a correlation between the objective value and the execution time. So, aumenting the number of instances results more accurated could be reached.

References

[1] Paul Erdös. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.

[2] Thomas Fruchterman and Edward Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

[3] Giorgio Gallo, Michael D Grigoriadis, and Robert E Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989.

[4] NetworkX. Source code for networkx.drawing.layout, 2018. `https://networkx.github.io/documentation/latest/_modules/networkx/drawing/layout.html`, Last accessed on 2019-03-31.

[5] Jyh-Jong Tsay, Bo-Liang Wu, and Yu-Sen Jeng. Hierarchically organized layout for visualization of biochemical pathways. *Artificial intelligence in medicine*, 48(2-3):107–117, 2010.