Power Grid Analysis in Apache Spark

Omar Mehio
American University of Beirut

Introduction

- Contingency analysis is a security function to assess the ability of a power grid to sustain various combinations of power grid component failures at energy control centers. Our project tends to explore the strength of the Lebanese Power Grid in substaining failiures.
- The power grid is a sparse graph. We apply 3 algorithms to exploit the network : Breadth First Search, Betweenness Centrality and Cascading Attack.
- The project handles distributed graph processing and Big Data.

Betweenness Centrality

- Betweenness Centrality measures the pairwise dependency of each vertex based on the following formula :
- $$g(v) = \Sigma_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$
- This equation states that the centrality score of a vertex v : is the ratio of the number of shortest paths from vertex s to vertex t which v lies on, to the total number of shortes paths from s to t.

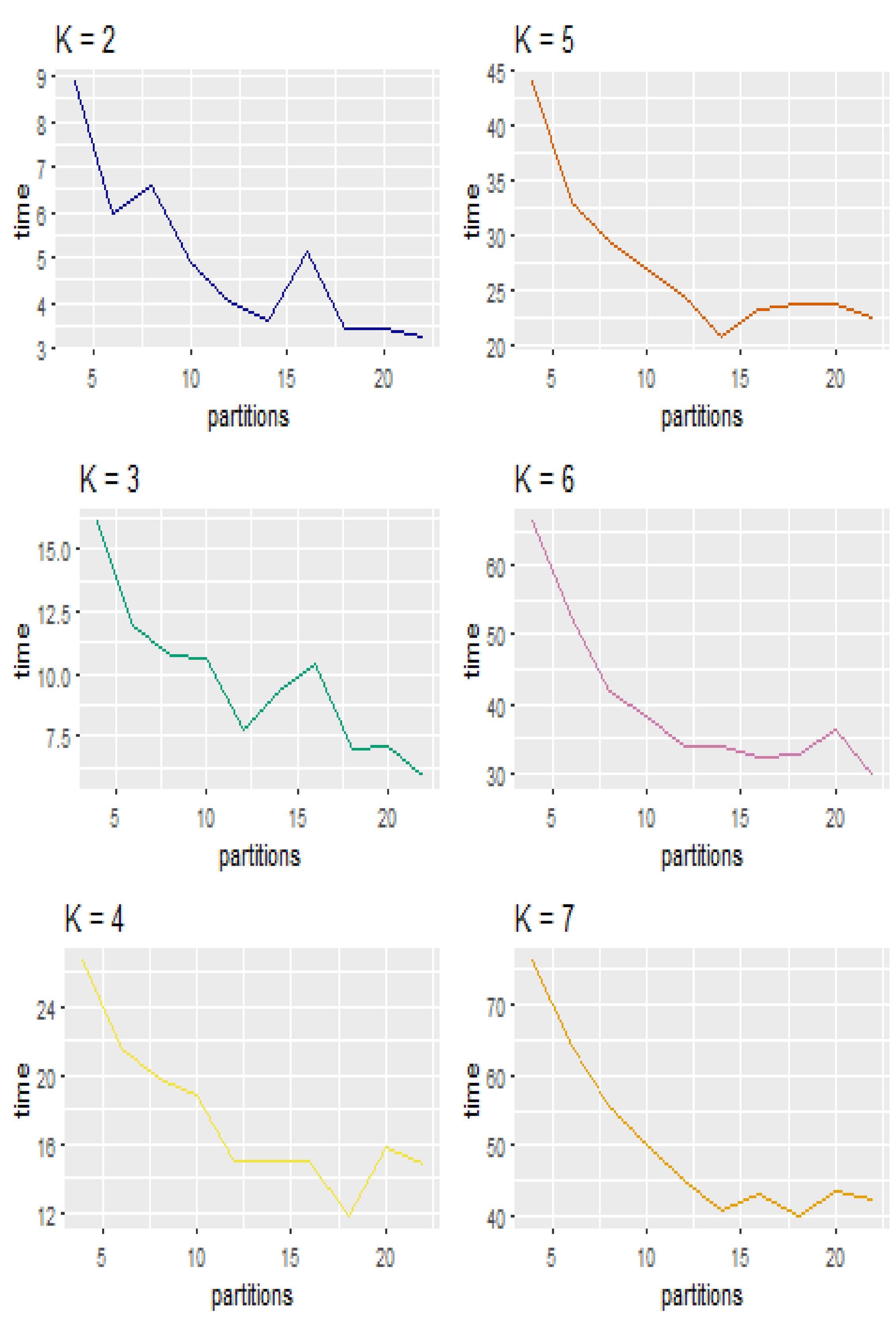


Figure 1: KBC Benchmarks against multiple partitions

Error Computation

- Since we are only diving k-steps we are introducting a computational error, since each vertex knows less about their neighbours.
- $$error = \frac{\Sigma_{v \in G(V)} |BC(v) - KBC(v)|}{V}$$
- In this formula, we implemented a serial version of Betweenness Centrality, thus we take each vertex in both KBC and the single threaded BC and we take the absolute value of their difference. Finally we aggregate these differences and divide by the total size of the graph.

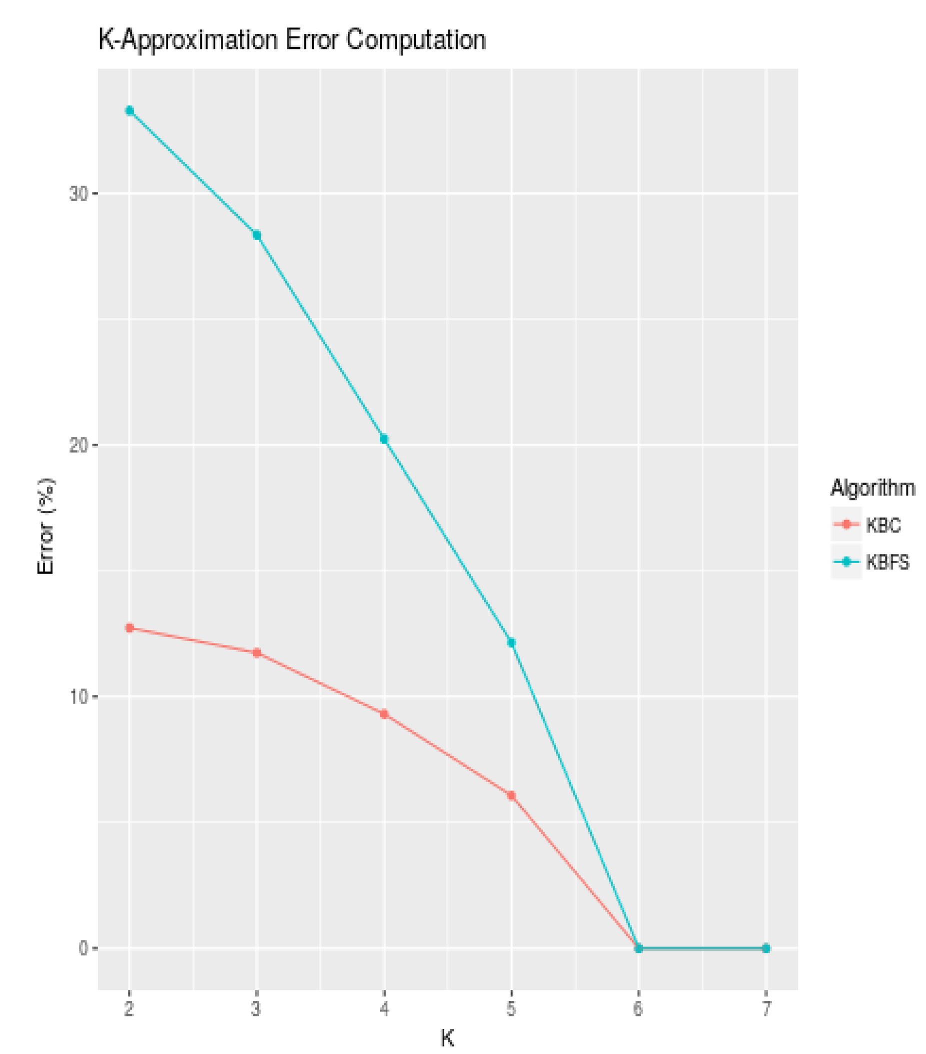


Figure 2: Error

Strongly Connected Components

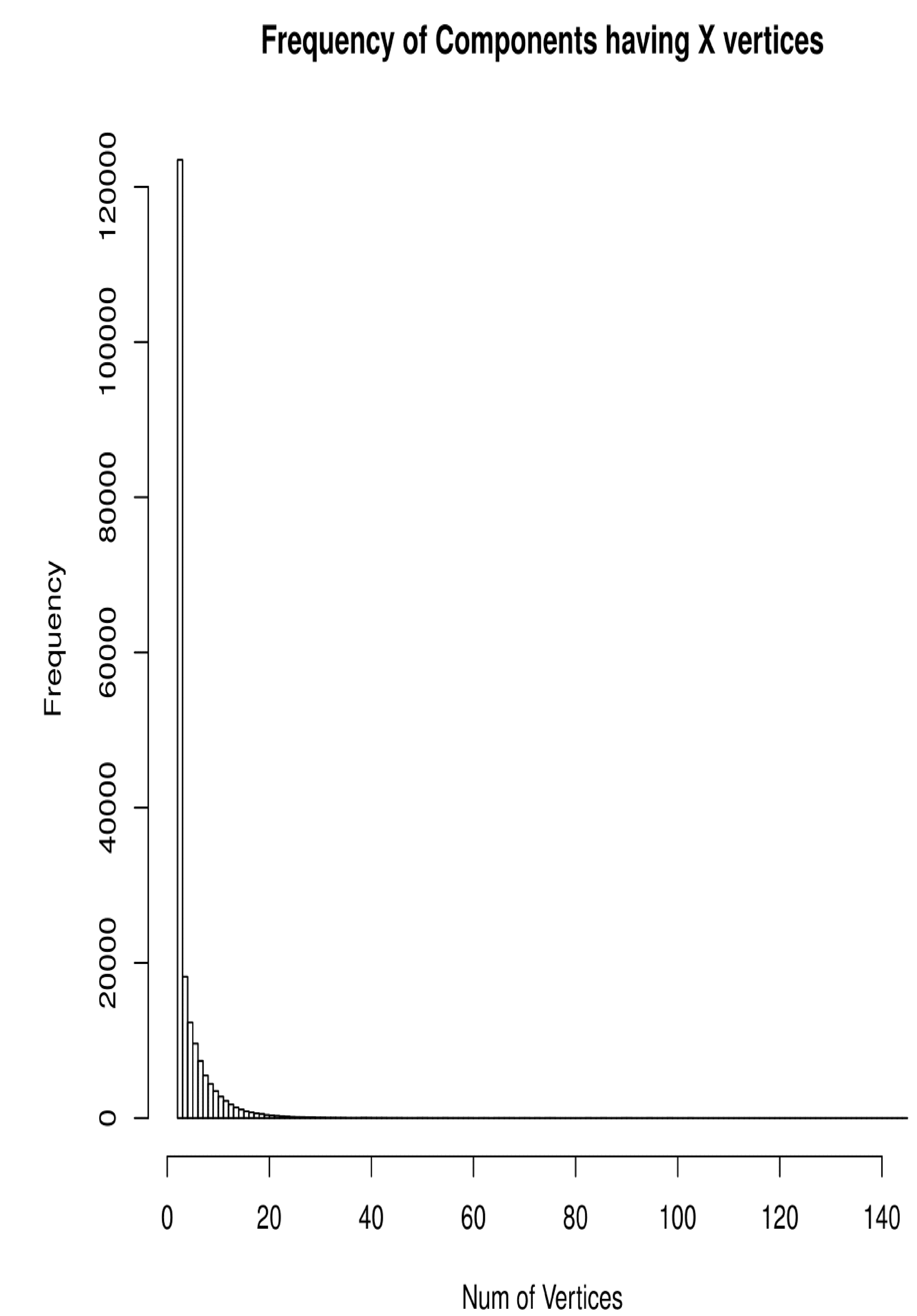
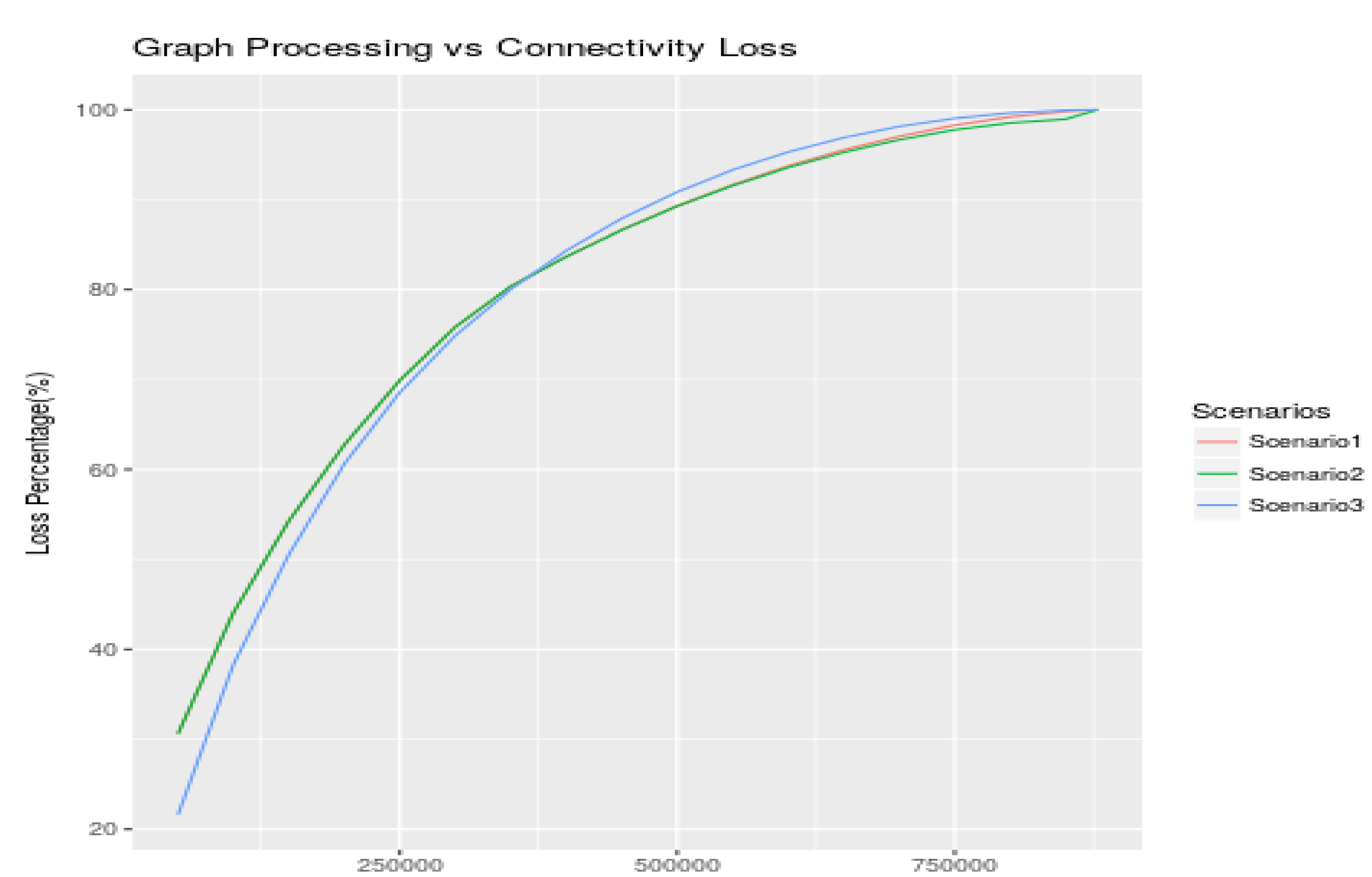


Figure 3: SCC

Result



Cascading Attack

Algorithm 1 Cascading Attack (G(V,E))

$total \leftarrow BFS(g)/KBFS(g,k)$
for $i = 1$ to N **do**
  $scores \leftarrow BC(g)/KBC(g,k)$
  $maximum \leftarrow scores.reduce(max)$
  $g \leftarrow g.subgraph(vertex! = maximum.getVertex)$
  $removed \leftarrow BFS(g)/KBFS(g,k)$
  $attacks[i] \leftarrow total - removed$
end for

Scenarios Benchmarks

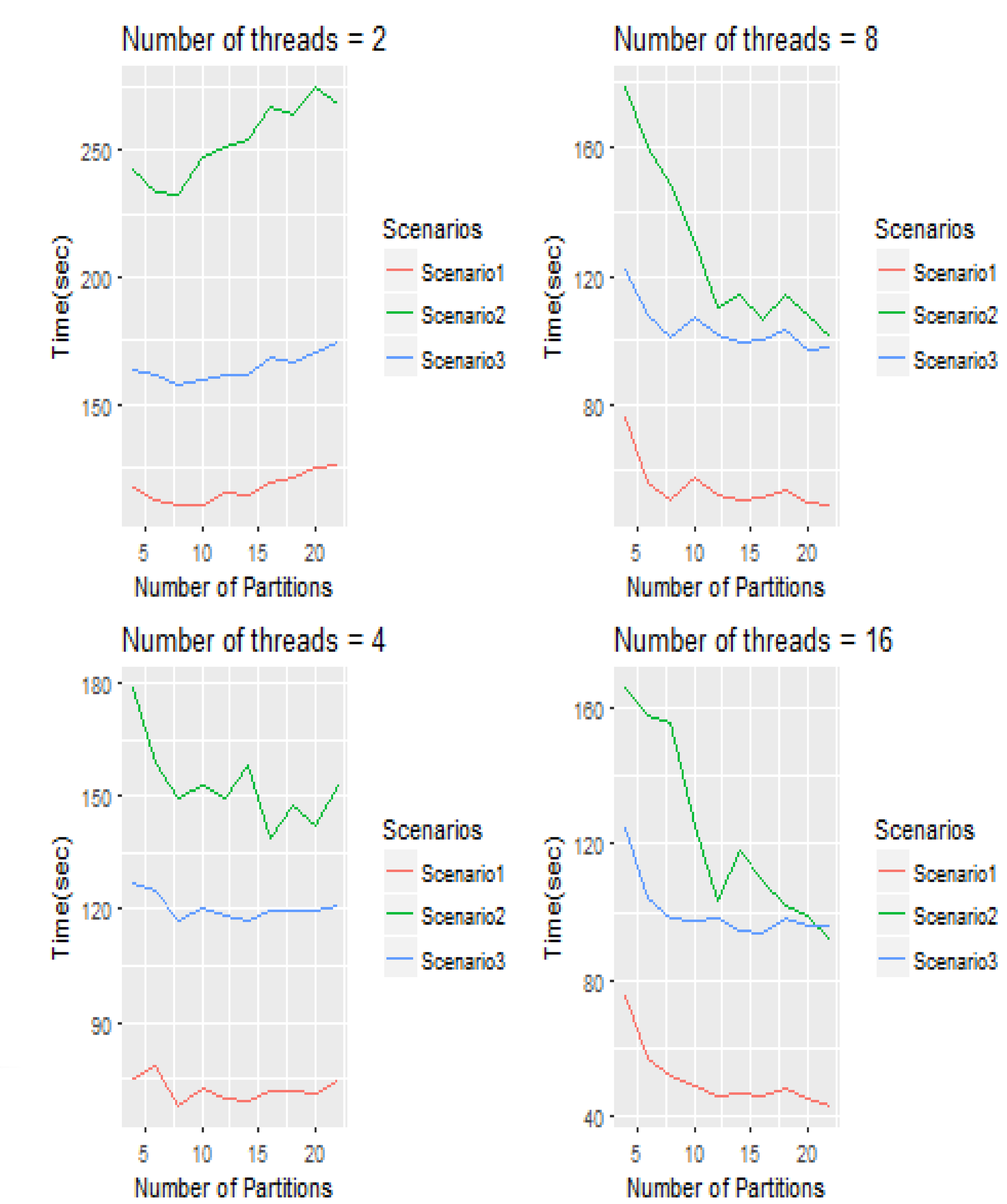


Figure 5: Scenarios Benchmarks against number of threads

References

[1] Ulrik Brandes *A Faster Algorithm in Betweenness Centrality.* University of Konstanz.

Contact Information

- Web: http://www.aub.edu.lb
- Email: okm02@mail.aub.edu