

Partitioning and Visualization of Large Graph

Swati A Bhavsar* and Varsha H Patil

Department of Computer Engineering, MCERC, Nashik, India

***Corresponding Author:** Swati A Bhavsar, Department of Computer Engineering, MCERC, Nashik, India.

Received: April 14, 2021

Published: May 24, 2021

© All rights are reserved by **Swati A Bhavsar and Varsha H Patil.**

Partitioning and visualization of large graph are very useful for analyzing complex data sets like social networking sites, experimental devices and sensor networks. A graph is a set of vertices and edges. It is a general data structure that models arbitrary relationship among objects, where vertices are objects and relationships as edges. A large graph is one which consists of hundreds to thousands of nodes and millions of edges. Some common examples of large graph include Very Large Scale Integrations (VLSI) designs, telephone networks, social networking sites, flight systems, Facebook, medical applications, publication records among many more. For VLSI designs, units on a chip are represented as vertices and wires connecting them as edges. In the case of social networking sites, vertices represent web pages and edges represent links among them. Similarly, for publication records, authors symbolize vertices and co-authors symbolize edges. DBLP (Digital Bibliography and Library Project) is a dataset of authors and co-authors relationships. Graph drawn as shown in figure 1a is an example of a large graph and figure 1b shows a large graph of the DBLP dataset.

Large graphs are sizably voluminous graphs and are commonly represented by means of popular data structures - adjacency matrix and adjacency lists. Memory Requirement for storage of matrix is too high as compared to adjacency list. Dynamic updating in nodes and edges are difficult operations especially in case of the adjacency matrix.

A large graph is an intricate data structure and it requires excessive processing, higher memory for storage and knowledge of patterns of the graph. To find patterns in a large graph, it is desirable to analyze, visualize, summarize and mine it. Some of the graphs that change with time are known as dynamic graphs. So it is very difficult to comment on the exact size and pattern of such

large and dynamic graphs. These large graphs cannot be analyzed and visualized as a whole. There is a need to partition the large graph into smaller subgraphs for effective analysis. Hence powerful graph partitioning along with graph visualization is to be used while processing and analyzing large graphs efficiently.

Figure 1: (a) Large graph in general. (b) Large graph for DBLP.

Large graphs are viewed using graph visualization techniques. Cutting the graph into small parts is one of the algorithmic operations is known as graph partitioning.

If $G = (V, E)$ is a graph, then graph partitioning is a process to divide vertex set V into k parts (subsets) v_1, v_2, \dots, v_k such that the subsets are disjoint and are of equal size, partitioning is balanced and the number of edges with endpoints in different subset is minimized.

For analysis of graph, at a given time, it is very important to partition the sizably voluminous graph into smaller subgraphs so that graph will fit into relatively less memory without minimum information loss in terms of connectivity. For partitioning a minimum number of edge cuts are desirable. Any edge that is cut in order to partition a graph into smaller graphs can be called as cut-edge. Therefore, finding the threshold value of an edge-cut plays a paramount role in the processing and analysis of large graphs.

The level of graph hierarchy by which graph can be seen is known as graph visualization. Graph visualization is a representation of interconnected nodes arranged in space and its representation in a particular structure so that user can understand it easily.

As the size of the graph is too large, it becomes one of the major challenges in efficient graph visualization. Large graph can cause various difficult problem named as - algorithmic complexity, display clutter, readability and navigation:

- **Algorithmic complexity:** Graph size is vital to algorithms in some cases because a lot of useful graph algorithms are either NP-complete or NP-hard. Therefore, some layout algorithms can be totally unusable when facing graphs of a large size. Algorithms require long processing time and make it hard to interact in real-time.
- **Display clutter:** Large Graph contains thousands of vertices or more. When the size of the data grows, graph becomes visually cluttered and confusing. It becomes difficult for user to discriminate between nodes and edges.
- **Readability:** Human perception able to visualize only small for graphs including firstly finding the node and secondly finding links between nodes.
- **Navigation:** Navigating large information spaces, such as graphs with thousands of nodes suffers from the problem of viewing a large space on a small display.

The large graph analysis is performed with the help of an efficient portioning and efficient visualization. The large graph problem can be treated through graph hierarchies, according to which a graph is recursively broken to define a tree of sets of partitions. For understanding a graph hierarchy, some of the important terms used are as below:

- **Hierarchical navigation:** It is the relation between arbitrary groups (partitions) of nodes.
- **Representation and processing:** Adjacencies of given graph nodes are computed in consideration with an entire graph, instead of a particular partition.
- **Mining:** From a given subset of nodes in the graph, particular induced subgraph that best summarizes the relationships of these subsets are to be identified efficiently.
- **Visualization:** The levels of the graph hierarchy that are can be seen.
- **Interaction:** It shows that how the above-mentioned tasks is efficiently and intuitively performed.

Volume 3 Issue 6 June 2021

© All rights are reserved by Swati A Bhavsar and Varsha H Patil.