Graph Clustering Using Multiway Ratio Cut

Tom Roxborough and Arunabha Sen*
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287
USA

Abstract. Identifying the *natural clusters* of nodes in a graph and treating them as *supernodes* or *metanodes* for a higher level graph (or an abstract graph) is a technique used for the reduction of visual complexity of graphs with a large number of nodes. In this paper we report on the implementation of a clustering algorithm based on the idea of *ratio cut*, a well known technique used for circuit partitioning in the VLSI domain. The algorithm is implemented in WINDOWS95/NT environment. The performance of the clustering algorithm on some large graphs obtained from the archives of Bell Laboratories is presented.

1 Introduction

Graphs are frequently used to model problems from various diverse domains such as telecommunication networks, VLSI circuit design, databases and computational chemistry. In these domains the nodes are used to represent certain entities of that domain and the edges represent the relationships between them. The relationship between the entities can be very effectively conveyed visually by a *nice* layout of the graph. However, in most of the realistic problem instances, the number of nodes and edges is far too many for a nice layout and also for comprehension of the information the graph was supposed to convey. In such situations, an abstract graph is constructed where each node represents a set of nodes of the original graph and the edges represent the relationship between these sets of nodes. Thus an abstract graph construction problem reduces to the problem of partitioning the node set of the original graph $G = (V, E)$, into a subset of nodes $V_1, V_2, ..., V_k$, such that $\cup_{i=1}^{k} V_i = V$ and $V_i \cap V_j = \emptyset$ for $i \neq j$.

The subsets V_is $(1 \leq i \leq k)$ are known as the *clusters* of the graph $G = (V, E)$. One problem with this approach is that there is no consensus among the researchers as to what constitutes a cluster. There is some intuitive understanding of what constitutes a cluster but there is no universally accepted formal definition of a cluster. In case the nodes and edges of the graph have some semantic information associated with them (in the form of labels), such information can be used for the purpose of clustering (or *grouping*) the nodes. An example of such

* Corresponding author, Telephone: 602-965-6153, Fax: 602-965-2751, e-mail: arunabha.sen@asu.edu. This work was partially supported by a grant from Tom Sawyer Software under NIST Advanced Technology Program.

information could be the IP addresses associated with the nodes in a telecommunication network. In case the graph has no such information, then the structural properties of the graph have to be utilized for the purpose of generating the clusters. We will refer to such graphs as the *flat graphs*. Several candidates for the structures have been proposed in the literature. These include *biconnected components* [?], *paths and triangles* [?], *circles of cliques* [?]. The reader is referred to [?] for discussions of some other possible structures for clustering. In spite of the differences of opinion as to what constitutes a cluster, one idea is universally accepted: the nodes belonging to a cluster must have a *strong relationship* between them in comparison with the nodes outside the cluster. In case of a flat graph this translates to finding a partition that minimizes the number of inter-cluster edges (or maximizes the intra-cluster edges).

2 Ratio Cut Technique

The Ratio Cut technique was proposed in [?] for the purpose of identifying the *natural clusters* of a graph. The technique was proposed in the VLSI domain for the circuit partitioning problem. Both in the case of circuit partitioning as well as graph clustering, minimization of the *cut edges* is a very important objective. In case the node set needs to be partitioned into only two subsets V_1 and V_2, the minimum cut partition can easily be computed using the *max-flow* techniques. However, the technique does not have any control on the size of subsets V_1 and V_2. In the VLSI domain each of the subsets has to fit into an integrated circuit chip and as such the size of each subset has to conform to some prespecified maximum limit. Therefore, the max-flow technique is not very useful in the circuit partitioning problem. The Kernighan-Lin technique, a well known heuristic for circuit partitioning, requires that the size of the two partitions V_1 and V_2 of the node set V be equal. This technique heuristically tries to find a partition with a small cut value, all the while keeping the size of the two subsets V_1 and V_2 equal.

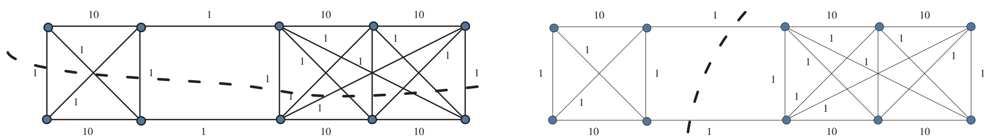

Fig. 1. Partitions produced by Kernighan-Lin Algorithm and Ratio Cut Algorithm respecively

As shown in [?], such a strict size requirement often forces a partition with a high cut size. The K-L technique produces the partition shown in figure 1, whereas a partition into natural clusters with a much better cut size is shown in figure 2. To attain the twin objectives of (i) minimizing the cut value and (ii) minimizing the difference in the size of the subsets, the authors of [?] proposed a new metric called the *ratio cut* to measure the quality of a partition. The ratio cut is defined as follows:

Consider a graph $G = (V, E)$. Suppose $c_{i,j}$ is the capacity (or the weight) of the edge connecting the nodes i and j. Suppose V_1, V_2 is a partition of the node set V. $(V_1 : V_2)$ denotes a cut that separates the nodes of V_1 from the nodes of $V_2 = V - V_1$. The capacity of this cut is equal to

$$C(V_1, V_2) = \sum_{i \in V_1} \sum_{j \in V_2} c_{ij}$$

The corresponding *ratio* value is given by

$$R(V_1, V_2) = C(V_1, V_2) / \mid V_1 \mid \times \mid V_2 \mid$$

The *ratio cut* is defined to be the cut that has the minimum ratio among all possible cuts of the graph, i.e., a cut (V_1, V_2) will be known as a ratio cut if

$$R(V_1, V_2) = \min_{X \subset V; Y = V \setminus X; X, Y \neq \emptyset} R(X, Y)$$

3 Implementation

As seen in the discussion in the previous section, the ratio cut technique proposed in [?] is applicable for a two-way partition of the graph. The authors dealt with the multi-way partition problem by repeated application of the two-way partition. For the graph clustering problem, we adapted the two-way ratio cut principle to a multi-way partition problem. In case of a k-way partition we compute the ratio as follows:

$$R(V_1, V_2, ..., V_k) = C(V_1, V_2, ..., V_k) / \mid V_1 \mid \times \mid V_2 \mid \times ... \times \mid V_k \mid$$

where

$$C(V_1, V_2, ..., V_k) = 1/2 \sum_{p=1}^{k} \sum_{i \in V_p} \sum_{j \notin V_p} c_{ij}$$

The *ratio cut* is defined exactly the same way as before, that is the cut that has the minimum ratio among all possible cuts of the graph.

In our implementation, we did not put a limit on the maximum number of nodes in a cluster, as we felt that such a restriction is artificial. However, our algorithm requires the user to specify the number of clusters in which the node set should be partitioned. The rational for this requirement is the following: The reason for clustering the nodes of the graph is to reduce the visual complexity and as such it should be left to the user to determine what level of complexity is acceptable for his application.

4 Experimental Results and Discussions

We tested our implementation of the multiway ratio cut based clustering algorithm with a wide range of graphs - small, medium and large. We created some example graphs for testing purposes and extensively used the Bell Laboratories graph library. This library has a large collection of graphs of wide range of variation in terms of number of nodes, edges and node degrees. Some representative examples of the output of our clustering algorithm is attached. The clustering algorithm runs on both the UNIX and the PC environment. We used Tom Sawyer Software's Graph Layout Toolkit for the layout of the graphs. The clustering algorithm computed the clusters in less than a few seconds in almost all of the Bell Laboratories graphs. Only in a small number of cases it required a few minutes for clustering.

5 Demo Environment

The clustering algorithm was implemented on a WINDOWS95/NT environment using Microsoft Visual C++ version 4.0. All graph layouts were produced by the Graph Layout Toolkit version 2.3 of Tom Sawyer Software Corporation. A PC with a 486 or Pentium processor along with 16MB of RAM is the only hardware requirement for the demo. The demo currently runs under either WINDOWS95 or NT.

6 Acknowledgements

The authors wish to acknowledge Jay Noh for his work on the initial development of the clustering technique.

References

1. R. Sablowski and A. Frick, "Automatic Graph Clustering," *Proceedings of Graph Drawing'96*, Berkeley, California, September,1996.
2. U. Dogrusoz, B. Madden and P. Madden, "Circular Layout in the Graph Layout Toolkit," *Proceedings of Graph Drawing'96*, Berkeley, California, September,1996.
3. P. Eades, "Multilevel Visualization of Clustered Graphs," *Proceedings of Graph Drawing'96*, Berkeley, California, September,1996.
4. Y. C. Wei and C. K. Cheng, "Ratio Cut Partitioning for Hierarchical Designs," *IEEE Transactions on Computer Aided Design*, vol. 10, no. 7, pp.911-921, July 1991.
5. C. J. Alpert and A. B. Kahng, "Recent Directions in Netlist Partitioning: A Survey," *INTEGRATION: the VLSI journal*, vol. 19, pp.1-81,1995.
6. D. Kimmelman, B. Leban, T. Roth, D. Zernik "Dynamic Graph Abstraction for Effective Software Visualization," *The Australian Computer Journal*, vol. 27, no. 4, pp.129-137, Nov 1995.
7. F. J. Brandenburg, M. Himsolt and K. Skodinis, "Graph Clustering: Circles of Cliques," submitted to *Graph Drawing'97*.

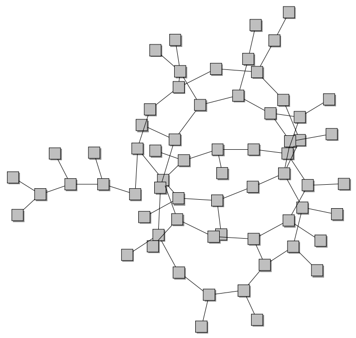

Fig. 2. Bell Lab graph 1572 before clustering

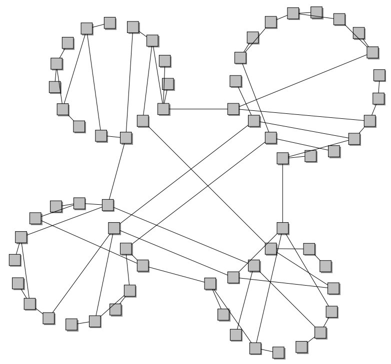

Fig. 3. Bell Lab graph 1572 with 4 clusters

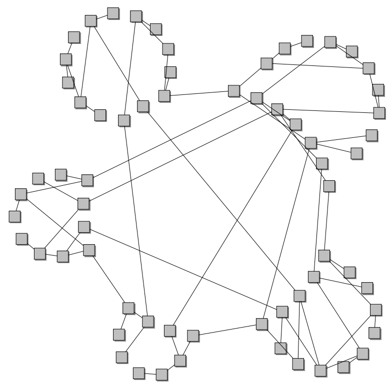

Fig. 4. Bell Lab graph 1572 with 5 clusters

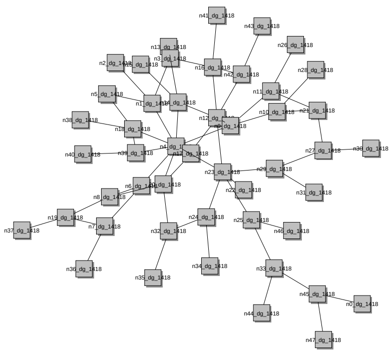

Fig. 5. Bell Lab graph 1487 before clustering

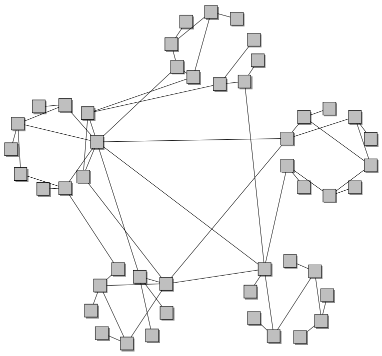

Fig. 6. Bell Lab graph 1487 with 5 clusters

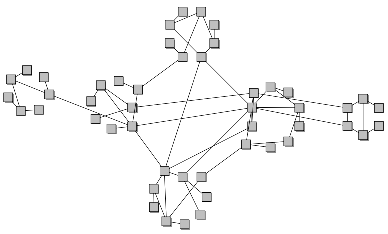

Fig. 7. Bell Lab graph 1487 with 6 clusters