Maxflow, min-cuts and multisectors of graphs

Cleve Ashcraft * Iain Duff †

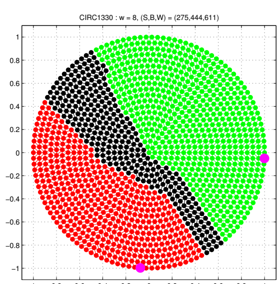
July 7, 2014

Abstract

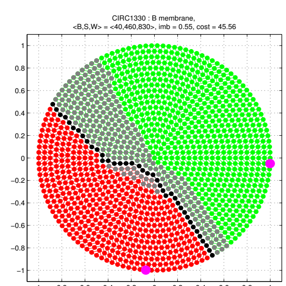
The vertex bisection problem can be addressed with matching and network flow techniques. Pothen and Fan [2] and Ashcraft and Liu [1] showed how matching, the Dulmage-Mendelsohn decomposition, and maxflow allowed one to find one or more minimal weight vertex separators chosen from a subset of vertices that form a wide separator.

For unit weight bipartite graphs, we can use matching and Dulmage-Mendelsohn. When the vertices do not have constant weight, or when the subgraph is not bipartite, we must solve maxflow over a network to find minimal weight separators.

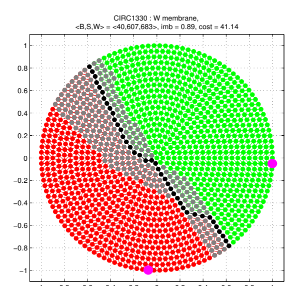
Here are the mechanics for vertex bisection. The set of candidate vertices form a wide separator, shown as black vertices in the leftmost figure below. There are two subdomains, shown as red and green. We associate a source node with the red subdomain and a sink node with the green subdomain and use these together with the vertices of the wide separator to construct a network on which we will run our maxflow algorithms.



Wide separator



Membrane B



Membrane W

Each vertex in the wide separator is identified with two nodes of the network that are joined by an arc in the network that has finite capacity. Other arcs, representing edges between wide separator vertices, or connecting the source and sink to vertices,

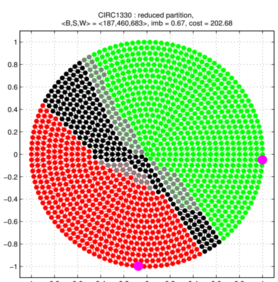
*Livermore Software Technology Corporation, 7374 Las Positas Road, Livermore, CA 94550. cleve@lstc.com

†STFC Rutherford Appleton Laboratory, Harwell Oxford, OX11 0QX UK and CERFACS, Toulouse, France. iain.duff@stfc.ac.uk

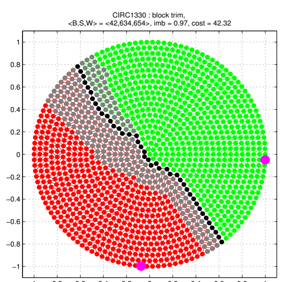
have infinite capacity. When we have found a maxflow through the network, any arc in a min-cut must have finite flow and will correspond to a vertex in the wide separator.

We conduct a search from the source to find a min-cut (middle figure above), and also from the sink (right figure above) to find a possibly different min-cut. Both induce a minimal weight vertex separator chosen from vertices in the wide separator. Each cut induces a partition, and we can choose the better of the two based on the partition with better balance.

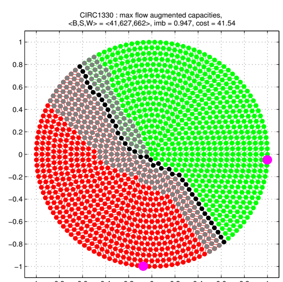
When the two min-cuts are identical the partition is unique. When the min-cuts are not identical, they trim the wide separator and we call these sets "membrane separators" each associated with a single subdomain. These membrane separators are minimal. If we move vertices from the wide separators to the domains, we have a reduced partition, where the new wide separator is smaller (or no larger) than the original. The left plot below shows the reduced wide separator where the vertices in grey will move into the two subdomains.



Reduced partition



Block trimming



Augmented capacities

We cannot use our maxflow algorithm again, there is no new information to be given by the two membrane separators. But we can use other algorithms. Here are two examples.

- Block trimming creates a minimal separator. Its run time is linear in the width of the wide separator.
- We augment the capacities to take into account the balance of the partition. Using these augmented capacities we solve maxflow and generate a second, further reduced partition. On the right we see that the reduced partition is actually minimal.

The issues become more interesting and the solutions less satisfactory when we consider three or more subdomains, where the separator is a multisector, not a bisector.

- Maxflow generates a membrane bisector around each subdomain.
- The union of the membrane bisectors may be a strict subset of the multisector, i.e., there may be vertices in the multisector adjacent to no domain.
- Maxflow followed by block trimming produces a minimal multisector.
- Maxflow with augmented capacities is also a good alternative, given a proper definition of augmenting for balance.

This is work in progress.

References

[1] Cleve Ashcraft and Joseph W. H. Liu. Applications of the Dulmage-Mendelsohn decomposition and network flow to graph bisection improvement. *SIAM Journal on Matrix Analysis and Applications*, 19(2):325–354, 1998.

[2] A. Pothen and C.-J. Fan. Computing the block triangular form of a sparse matrix. *ACM Transactions on Mathematical Software*, 16:303–324, 1990.