15-853: Algorithms in the "Real World" **Lecturer: Prof. Guy Blelloch**
Lecture 5, September 25, 2002 **Scribe: Barbara Anthony**

1 Graph Separators

We start by discussing the notion of the two kinds of graph separators, which we will formalize later. We are interested in graph separators partially because they are hugely practical in a great variety of applications, some of which we will discuss. The following topics are included in this lecture.

- Vertex and Edge Separators
- Types of graphs that have good separators
- Types of graphs that do not have good separators
- Applications of separators
- Algorithms for finding separators
- Preliminaries for Lipton-Tarjan Planar Graph Separators

1.1 Vertex and Edge Separators

Given a graph $G = (V, E)$, we consider two kinds of separators, namely edge separators and vertex (node) separators.

In edge separators, illustrated in Figure 1, we partition the nodes into two sets, and we want to minimize the number of *cut edges*, which is the number of edges between the two sets of nodes, here denoted by C. We also want to maintain some sort of balance between the sets, which we also do not formalize at this time. However, intuitively we know that this forbids the uninteresting case of making the sets be a node v of minimum degree and $V \setminus v$. The strictest interpretations may require that the sets be perfectly balanced, i.e. that they each contain the same number of nodes.

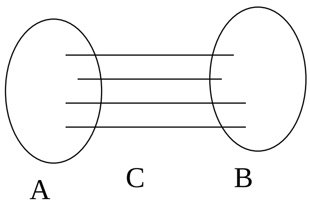

Figure 1: Edge Separator

In vertex (node) separators, illustrated in Figure 2, we partition the nodes V into three sets of nodes A, B, and C, where there are no edges between vertices in A and B. The goal is to minimize the number of nodes in C, again maintaining an appropriate balance between A and B.

Separators are also known by several other names, including bisectors, bifurcators, balanced cuts, and partitions.

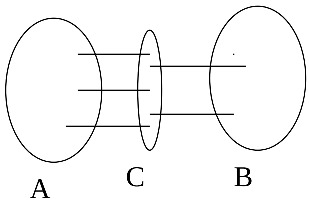

Figure 2: Vertex Separator

Note that for the purpose of finding good graph separators, directed graphs are treated as undirected graphs.

Claim 1.1. *A good edge separator for a graph G automatically provides a good vertex separator for G, provided that the balance restrictions are not too strict.*

Proof. Consider a graph G that has a good edge separator, with sets A and B. Choose one of these sets, and from it, construct C by choosing the vertices in the set that are endpoints of cut edges of the good edge separator. Since there can be at most one vertex in a set per cut edge, the vertex separator C must have size at most that of the edge separator, so it is good size-wise, given that the edge separator is good. Balance will be altered by the removal of the up to $|C|$ nodes from one of the sets, but as we have shown that C is relatively small, as it is a good separator, then provided that the balance restrictions are not too strict, reasonable balance is maintained.

■

Observe that the converse is false. The analogous method of obtaining an edge separator from a vertex separator by combining those vertices in C with either A or B does not guarantee a good edge separator. In particular, while C must be small if it is a good vertex separator, there could be a large number of edges out of C. Thus, in general, obtaining an edge separator from a vertex separator is hard.

For example, there are some types of planar graphs which have good vertex separators but do not have good edge separators. In particular, consider a star graph, as in Figure 3. Using the middle vertex as C and randomly partitioning the remaining nodes into A and B gives us a vertex separator of size 1. However, an edge separator will be of size $O(n)$ no matter how we partition the graph.

However, graphs with bounded degree, those graphs where the maximum degree of any vertex is less than some fixed constant, are instances where we can easily obtain a good edge separator from a good vertex separator. The problem that arose before in this creation was that we could have a large number of edges out of C. If the graph is of bounded degree Δ, then we know that there are at most $|C|\Delta$ edges out of C. Thus, we can merge C into either A or B, whichever it shares more edges with, and we have a good edge separator for the graph. Thus, the definitions of edge separator and vertex separator are equivalent within a constant factor for bounded degree graphs.

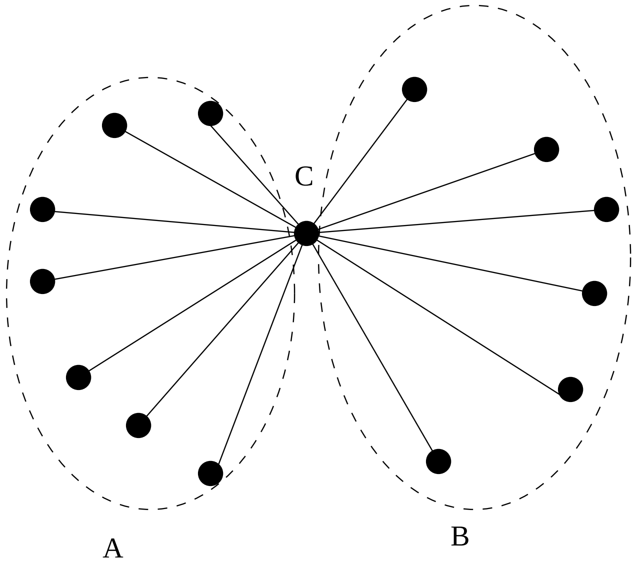

Figure 3: Good Vertex Separator for a Star Graph

1.2 Types of graphs that have good separators

When we consider types of graphs that have good separators, we mean those that have good recursive separators. In general we do not want to separate a graph only once, but rather we want to find a separator and apply the method recusively to the sets. This is useful in divide and conquer techniques and parallel processing algorithms.

Planar graphs, as well as degenerate forms such as trees, linked lists, and edgeless graphs, all have good separators. They arise in 2-D meshes and graphics applications. Here we can find separators of $O(\sqrt{n})$. Graphs of bounded genus and also some graphs with forbidden minors fit into this category.

A second set of graphs which have good separators are those which we will characterize as "almost planar" graphs, which can usually be made planar by removing a small number of edges. Some natural examples are road networks (made non-planar by bridges and tunnels), utility distribution networks, flight connections, and the internet. Note that in the case of the internet, a graph separator is typically much smaller than $O(\sqrt{n})$ because of the trunk lines.

Circuits, though a priori it may not be obvious, do have good separators. As one of the earliest applications for separators, they are the motivation for the work done in the 1960s by Kernighan and Lin, whose heuristic is mentioned later. As an aside, Charles Leiserson, the advisor of Dr. Blelloch and Dr. Maggs, developed a theory relating the amount of space upon which you can fit a circuit if the size of the wires must be included. As chips are designed in chunks, graph separators are useful for separating out the circuits across multiple boards, as we would like to minimize the number of wires between boards. Today VLSI design still makes use of this when we try to minimize the area used.

Social networks, somewhat less surprisingly, also have good separators. Instances of social networks include a "friends" graph, and the bibliographic references or citations on papers. Others include the link structure of the web (which, in fact, has very small separators), and even huge graphs such as those representing telephone calls.

Graphs embedded in low dimensions also have small separators. For example, with 3-D meshes and certain nicety conditions, we have a separator of $O(n^{\frac{2}{3}})$. It is not too difficult to verify this for the three-dimensional cube. In "unstructured meshes" with certain smoothness constraints, we get the same type of results, which are very common in finite element simulations. Generalizing, we get separators of order $O(n^{\frac{d-1}{d}})$. Note that when $d = 2$, this is precisely the $O(\sqrt{n})$ size mentioned earlier. Note that in low-dimensional space, we handle crossings in terms of ball systems, where each vertex has a ball surrounding it to its nearest neighbor, and vertices can only connect when balls overlap.

Fat trees, which are also associated with Leiserson, also have good separators. In a fat tree, to avoid the bottlenecking at the root associated with ordinary trees, the fat tree adds more and more links as we get close and closer to the root. A minimum embedding of an arbitrary graph with small separators can be done in a fat tree. These are used in various parallel machines.

1.3 Graphs that do not have good separators

Though the first thought might be that we would need graphs of high degree to not be able to find a good separator, note the following examples with average degree at most $O(\log n)$.

Hypercubes are good networks for designing parallel machines, precisely because they do not have good separators. An edge separator of size less than $O(n)$ cannot be found, nor can a vertex separator of less than $O(\frac{n}{\sqrt{\log n}})$. Dr. Maggs noted that it was somewhat surprising that the gap between the two is not the degree of the hypercube.

Expander graphs, discussed in the previous lecture, also do not have good expanders. Here we alter the definition of expander graphs to deal with graphs that are not bipartite by saying that we can look at any set of the appropriate size and see how many edges go to vertices outside the set. In particular, consider an expander graph with $\alpha = \frac{1}{2}$ and $\beta = 1.1$. Thus, for any half of the vertices, we can reach $1.1\frac{n}{2}$ vertices. Thus, it cannot have a small separator. Whatever set we are in can have only $\frac{n}{2}$ vertices, so there must be $.1\frac{n}{2}$ vertices in the other set that we can reach, so it has no good separator. This property makes such graphs useful for rapid mixing, as when we take a random walk on such a graph, we rapidly find ourselves at any vertex with equal probability.

Butterfly networks, or FFT (Fast Fourier Transform) networks have edge separators of size $O(\frac{n}{\log n})$. Since they are of bounded degree, they also have vertex separators of the same size. Consider a random graph with n vertices, m edges, and $m \geq 2n$. Most do not have small separators. However, most graphs that we encounter in practice are not actually random graphs, but are very structured.

2 Applications of Separators

There are numerous applications of separators which motivate some of the research regarding them. We have already discussed circuit layout on chips or boards as one of the original applications in the 1960s. They are also useful for partitioning data on parallel machines. For example, if we want to simulate a mesh but minimize the communication between processors, we could make use of graph separators. Another similar application is out of core algorithms, where we need to read back from the disk in chunks. Building layout can also be done with graph separators, where we can think of the placement of rooms as a 3-D analog to circuit layout.

These applications can be aided by the use of the METIS package, a software tool. The METIS package can also use graph separators for solving linear systems. Normally this is an $O(n^3)$ operation, but with planar graph separators of size $O\sqrt{n}$, it can be done in $O(n^{3/2})$ time. For 3-D meshes, with separators of size $O(n^{2/3})$, this becomes $O(n^2)$ time.

Graph separators also can provide better approximations to NP-hard problems, at lower costs. In particular, the approximation ratio for maximal independent sets decreases from 2 to something near 1 with the use of graph separators. There are also improvements in other problems, such as the traveling salesman problem.

Machine learning, and in particular, clustering algorithms can be aided by the use of graph separators. Highly clustered data represented in a graph theoretical model can be shown to have good separators, which can speed up the finding of clusters dramatically. Thus we can consider either the mathematical interpretation or the AI interpretation, and benefit from good separators.

The runtime of shortest path algorithms can be improved with graph separators. In particular, for single-source shortest path in planar graphs, the runtime can improve from $O(n \log n)$ to $O(n)$. And for the all-pairs shortest path problem, the runtime decreases from roughly $O(n^3)$ to $O(n^2)$. These problems have fast dynamic versions. In general, better bounds can be found for all sorts of flow and matching problems. And graph compression, which for planar graphs is normally $O(n \log n)$, is improved to $O(n)$ bits.

3 Algorithms for finding good separators

Finding an "optimal" graph separator is NP-hard. Again, we do not make precise what optimal means in this instance, as there are many vairants on what it means for a graph separator to be balanced. Some such variants include requiring half of the vertices in each partition, requiring no more than $\frac{2}{3}$ of the vertices on one side, or any other constant fraction. Others consider the min cut, which is minimize $\frac{|C|}{|A||B|}$. Still others deal with k-partitioning, where we are splitting the vertices into k sets. There are also weighted versions, where the vertices or edges can be weighted. Then we can attempt to minimize the weight of the cut, which is the sum of the weights of the cut edges, or to balance the partition weight, where the weight of a partition is the sum of the weights of the vertices it includes.

However, many of the applications that make use of graph separators require more than the knowledge that a good separator exists, but they also need to find small separators. Fortunately, finding a good reasonable separator is not to hard, depending upon the type of the graph. Though the variants mentioned and others are for the most part NP-hard, most are known to have reasonable approximations. In planar graphs, for instance, we can find separators that are of size $O(\sqrt{n})$ in linear time. Note however that if for example the graph has a separator of size $O(n^{1/4})$, we are not guaranteed to find it.

Finding good graph separators is an area of some research interest right now. In particular, it is one of the areas where theory and practice do not currently match, and thus we would like to see improvements made in what is known.

A 1970 paper of Kernighan and Lin entitled "An efficient heuristic procedure for partitioning graphs", [2] gives one heuristic, which uses local refinement. Work on planar graphs published in 1961 proved separators of size $O(\sqrt{n}\log(n))$. In 1979, Lipton and Tarjan [4] were able to prove size $O(\sqrt{n})$ with linear runtime. We will consider their algorithm later. Geometric approaches have also been considered, and it has been found that there are spherical cuts in 3-D that give good separators for well shaped graphs. In particular, a CMU researcher, Gary Miller showed in 1990 that for 3-D graphs this technique can find separators of size $O(n^{2/3})$.

Spectral approaches have also been used, where you take the Laplacian form of the adjacency matrix and find the second eigenvector. Then it gives a value on each vertex, so you can sort the vertices and split them in half. This procedure, for sound theoretical reasons, gives good separators under certain nicety conditions. It remains an open problem as to how broadly this works.

Leighton and Rao published work regarding flow-based techniques for finding separators, entitled "An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms" [3]. It should be noted that they use a different notion of balanced than we have previously mentioned. In particular, they consider the minimum quotient separator, which is given by minimize $\frac{|C|}{\min(|A|,|B|)}$. This is a way of balancing the tradeoff between the cut size and the amount of balance. Their results allow for the finding of a separator within a $\log n$ factor of optimal. However, this is not yet used in practice because, although it is poly-time, it is too expensive, since it uses multiple commodity flow as a subroutine.

The most popular technique in practice, used in software such as METIS, CHOCO, and Gnu packages, is multilevel recursive bisection, which will be discussed in a later lecture. A relevant paper is that of Karypis and Kumar [1] which presents a faster variant of the Kernightan and Lin algorithm and uses a multilevel scheme. While not much is known about this method theoretically, and a better understanding of when it works would be valuable, most packages switched to this method around five years ago from the spectral approach, because the spectral approach required the computation of eigenvectors, which is not cheap.

3.1 Planar Graph Separators

We will begin with some of the necessary definitions today in order to discuss in greater detail next time the Lipton-Tarjan algorithm for planar graph separators. To keep the discussion more concise, we skip one of the messiest sub-components (covered in 15-750). Note that what we provide is both an algorithm and a proof. In fact, the only proof that is widely known, perhaps the only kind of proof that there is, for proving that planar graphs have small separators is constructive in nature.

Definition 3.1. A *planar graph* is a graph that can be embedded in the plane (or in a sphere) with no crossings.

S is a class of graphs that is closed under the operation of taking a subgraph, so any subgraph must also be in the set. This is also called the hereditary condition.

Planar graphs are such a class of graphs, since if we remove any edge or vertex (and thus its associated edges) from a planar graph, the resultant graph is still planar. This property, however, is not unique to planar graphs – it holds for many other types of graphs.

Definition 3.2. S satisfies an $f(n)$-separator theorem if there are constants $\alpha < 1$ and $\beta > 0$ so that for every $G \in S$, with $|G|$ denoting the number of nodes in G, there exists a cut C with

- $|C| \leq \beta f(|G|)$
- $|A| \leq \alpha |G|$, $|B| \leq \alpha |G|$

Note that the first condition requires that the cut size be small, and the second condition guarantees a type of balance requirement. Observe also that the definition we provided is polymorphic, as it applies equally well for edge separators and vertex separators.

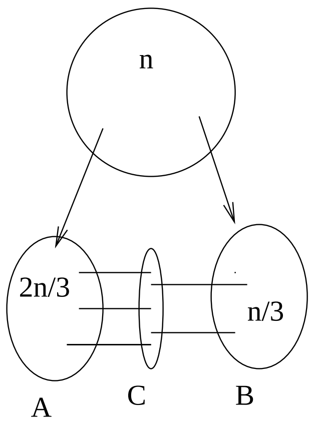

Figure 4: Split for a planar graph vertex separator

For planar graphs, we will show in the next lecture that they satisfy a $\sqrt{n}$-separator theorem, with constants $\alpha = \frac{2}{3}$ and $\beta = 4$. We thus obtain a split of the type illustrated in Figure 4.

References

[1] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. In *SIAM Journal on Scientific Computing*, Volume 20, Number 1, pages 359-392, 1998.

[2] Brian W. Kernighan and S. Lin, An efficient heuristic procedure for partitioning graphs. In *Bell Systems Technical Journal*, 49(2):291–308, 1970.

[3] F.T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with application to approximation algorithms. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, pages 422–431, 1988.

[4] R.J. Lipton and R.E. Tarjan. A separator theorem for planar graphs. In *SIAM Journal of Applied Math.*, 36, pages 177-189, 1979.