Topics in Algorithms 2005

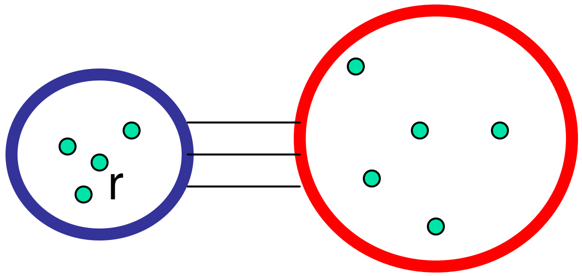
Disjoint Paths, Cuts, Arborescences

Ramesh Hariharan

Undirected Unweighted Cuts

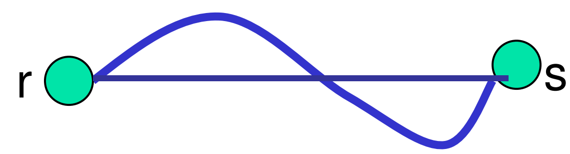
- Cut: Partition of the vertex set

- Cut Size: Number of edges between the two sides of the partition



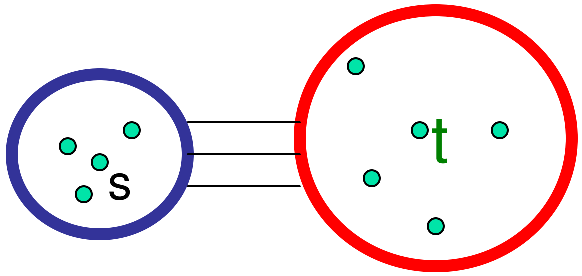
Undirected Unweighted Cuts

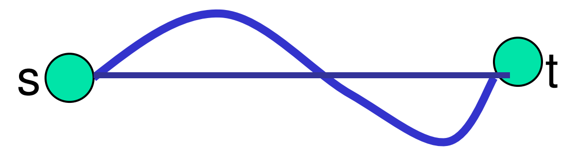
- Edge Connectivity between s and t: The number of edge-disjoint paths between s and t



Menger's Theorem

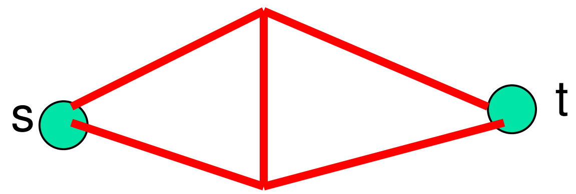
- The size of the smallest cut separating r and s equals the number of edge disjoint paths between s and t





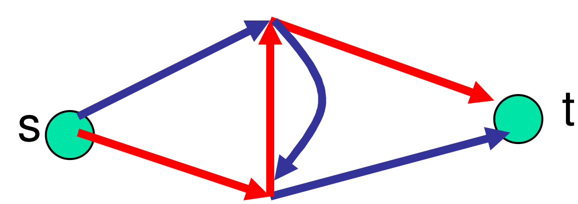
Proving Menger's Theorem

- Find paths from s to t repeatedly
- Stop when no more paths can be found
- Does this work?



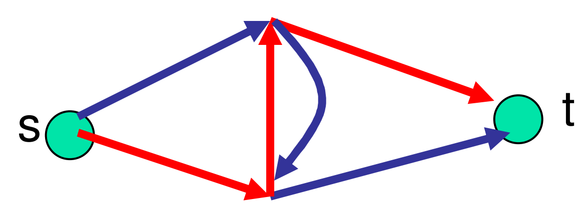
Proving Menger's Theorem

- Need to allow undoing of wrong decisions
- The Trick: Replace undir edge by 2 dir edges
  - An edge used in one direction is marked; it can still be used but only in the other direction
  - An edge used in both directions is unmarked; it behaves as if it was never used



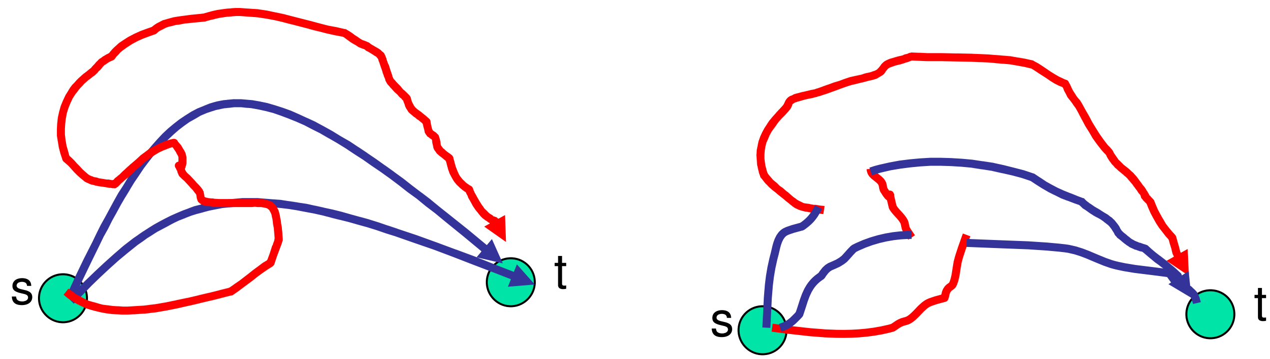
Proving Menger's Theorem

- Find directed paths repeatedly
- A path found must respect marking, i.e., it uses marked edges only in the unique available direction; unmarked edges can go in both directions
- From these directed paths, how do we get our undirected paths?
  - Problem: each undirected edge needs to be used exactly one over all paths; however, several directed paths may use a particular undirected edge



Proving Menger's Theorem

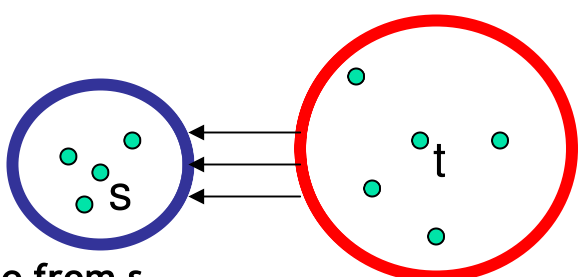
- By induction
- Suppose one has k paths
- Assume you can convert k-1 from directed to "undirected" maintaining the following
  - Edges used en even number of times amongst these k-1 paths are not used at all by these paths. Edges used an odd number of times are used exactly once and only in the "right" direction
- Now take the kth directed path and throw it in as shown below: REMEMBER to maintain the odd/even criterion above



Proving Menger's Theorem

- Fine, so we have as many undirected paths as directed paths
- But why do we have sufficiently many directed paths (as many as the min-cut)?
- Suppose not: so you run into a deadend, can't find any more paths
- Now demonstrate a cut of size as much as the number of paths found; how?

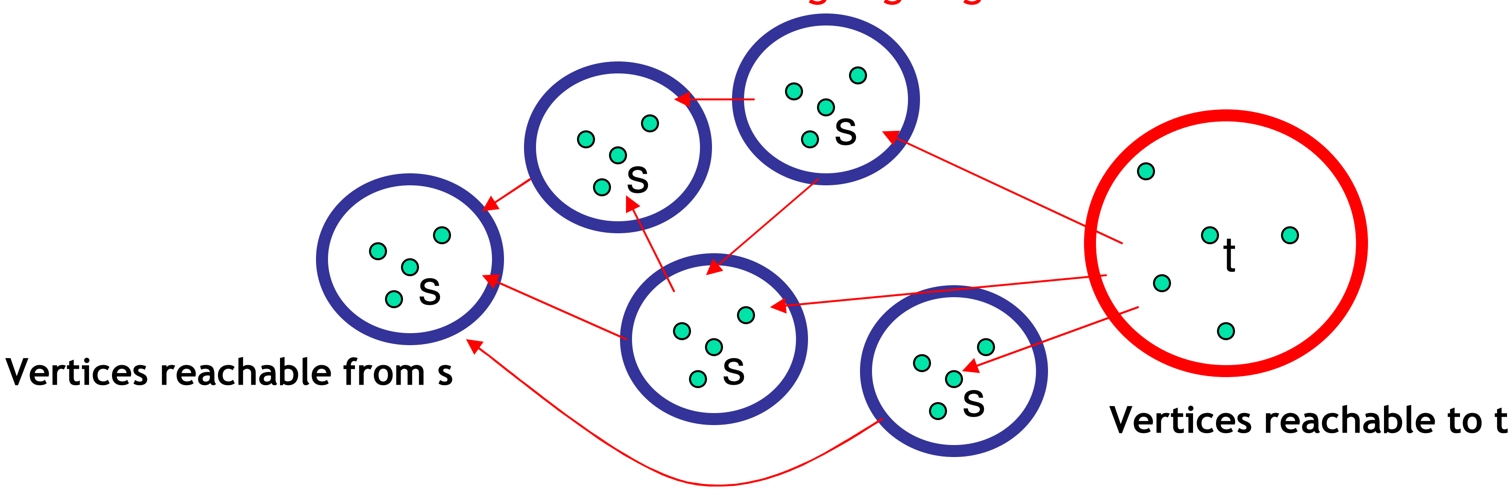
  - Take all vertices reachable from s in the unmarked graph
  - Claim: This is a cut of size equal to the number of paths found so far
  - Why?? Exercise



Vertices reachable from s

Capturing All s-t Min-Cuts

- The residual graph (comprising unused directed edges; if unmarked both directions are unused, otherwise only one direction is unsued) has components
- Vertices reachable from s
- Vertices reachable to t
- Strongly connected components comprising vertices other than s or t
- The residual graph can be thought of as a DAG on these components
- Any s-t cut in this DAG with all edges going towards s is a mincut: Exercise??
- All s-t cuts are mincuts in this DAG with edges going towards s: Exercise??

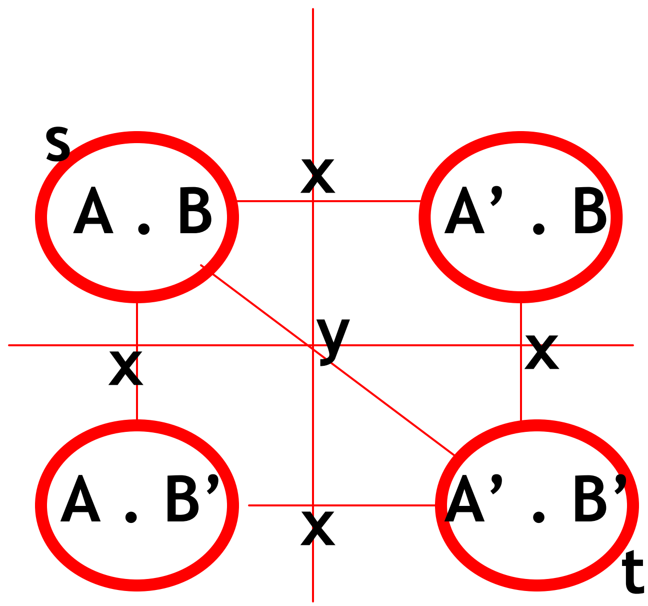


How many s-t min-cuts can be there?

- Take n edges organized into k disjoint paths of length a,b,c.. adding up to n
- How many cuts does this give??
- Maximize over a,b,c adding up to n

s-t cuts Nesting

- If you have two s-t min cut, show that their intersection is also a s-t mincut: Exercise??

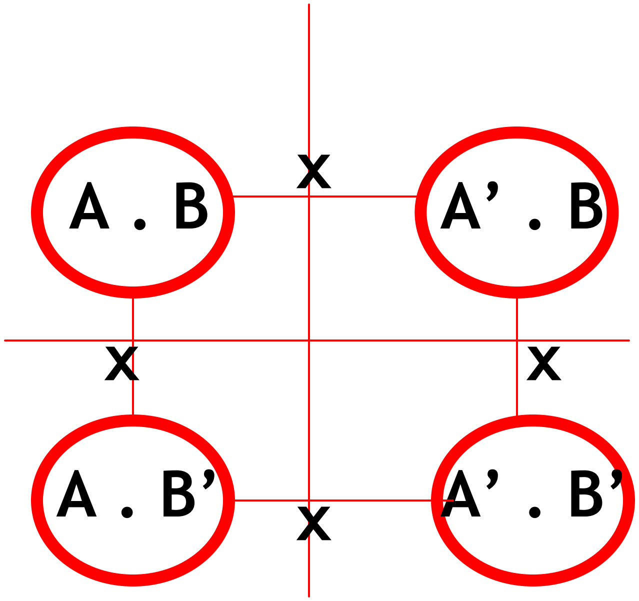


Global Min-Cuts

- Take s-t min-cuts over all s,t pairs;
- Cuts of the smallest size are called global min-cuts
- How many global min-cuts are there?

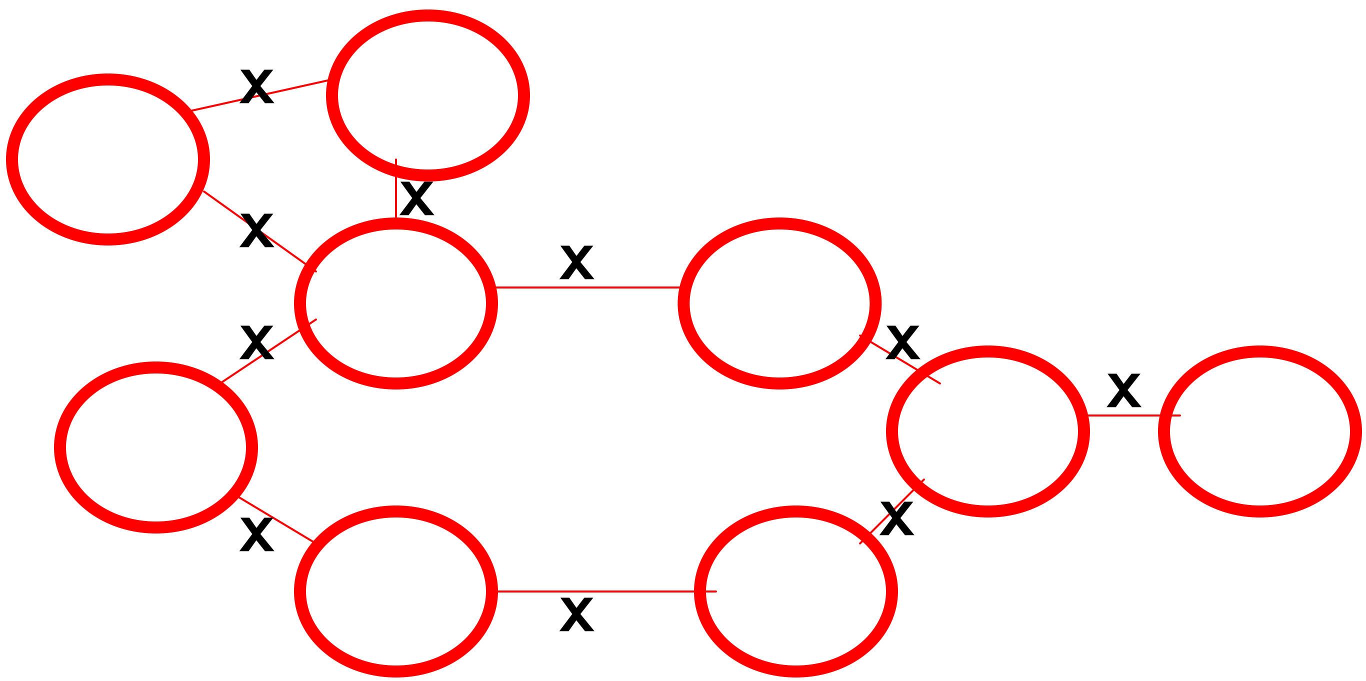
Global Min-Cuts Nesting

- If you have two global min cuts, show that all 4 cuts below are global min-cuts. Exercise??



The Cactus of Global Cuts

- All global min-cuts form a cactus. Any disconnection of the cactus is a min-cut. Exercise?? How many global min-cuts does this imply.



Finding a global min-cut

- A simple randomized algorithm by Karger
  - Take a random edge and contract it
  - What is the probability that you are left with a min-cut at the end
  - So no edge of the min-cut should get contracted
  - Prob of this hapenning in the first round is 1-k/e > (n-2)/n. Why??
  - Overall probability of success = 1/(n choose 2)
  - Repeat n^2 times for success
  - Time taken: O(n^2 * mlog n)
- Implementing one phase
  - Take a random permutation of the edges
  - Run what??

Edmonds Arborescences

- Analogues of Menger's Theorem for Global Connectivity
- Take an arbitrary s as root
- If global min-cut is k, then there are k edge disjoint arborescences.

- Arboresence: Spanning tree with all edges directed away from the root; as before we need to direct the graph by replacing edges in both directions.

- Example: take a simple cycle

Edmonds Arborescences: Proof

- Existence Proof: Next Class
- Open Problem: Construct these arborescences in sub quadratic time, even for the case k=2

References

- Papers by David Karger on Min-Cuts, available on the web
- Edmonds Theorem, Cactus Structures: Still trying to find good references, will provide next time

Thank You