

Minimum cuts (MinCut algorithm)

Calculates minimum cut in an undirected graph.

Definition of a cut:

Given a graph $G = (V, E)$ and a subset S of V , the cut $\delta(S)$ induced by S is the set of edges

$(i, j) \in E$ such that $|\{i, j\} \cap S| = 1$; i.e. $\delta(S)$ consists of all those edges with exactly one endpoint in S .

Each edge of an undirected graph has associated with it a non-negative cost or capacity c_e ; the cost of the cut $\delta(S)$, is the sum of the costs of the edges in the cut, i.e.

$$c(\delta(S)) = \sum_{e \in \delta(S)} c_e$$

The minimum cut problem is to find the cut of minimum cost. The algorithm considered here is a greedy algorithm.

Algorithm:

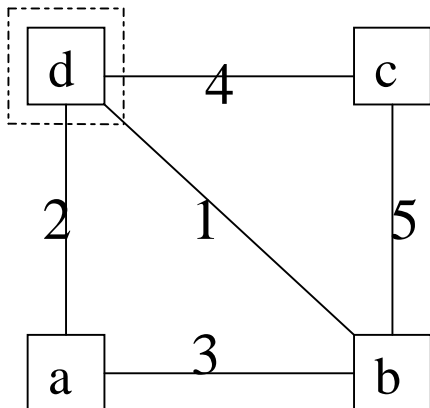
- Select a vertex of the graph.
- Build an ordering of vertices by adding to the selected vertex/vertices the vertex whose cost to the previous vertex/vertices is maximized.
- The cut induced by the last vertex is considered, as well as the cuts obtained by

recursively applying the method to the graph obtained by shrinking the last two vertices.

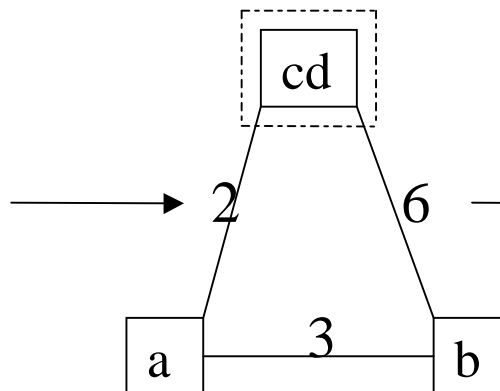
- If there are edges from a vertex v to the last two vertices then substitute those two edges with one edge having capacity equal to the sum of capacities of the two edges.
- The best cut among all possible cuts considered is the overall mincut.

Example:

cut = 7



cut=8



cut=5

