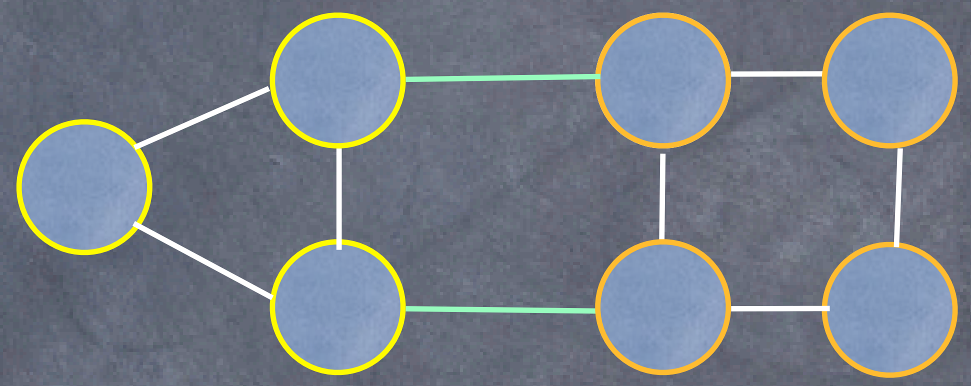
A Randomized Algorithms for Minimum Cuts

Andreas Klappenecker

Minimum Cut

A cut in a graph $G=(V,E)$ is a partition of the set V of vertices into two disjoint sets V_1 and V_2.



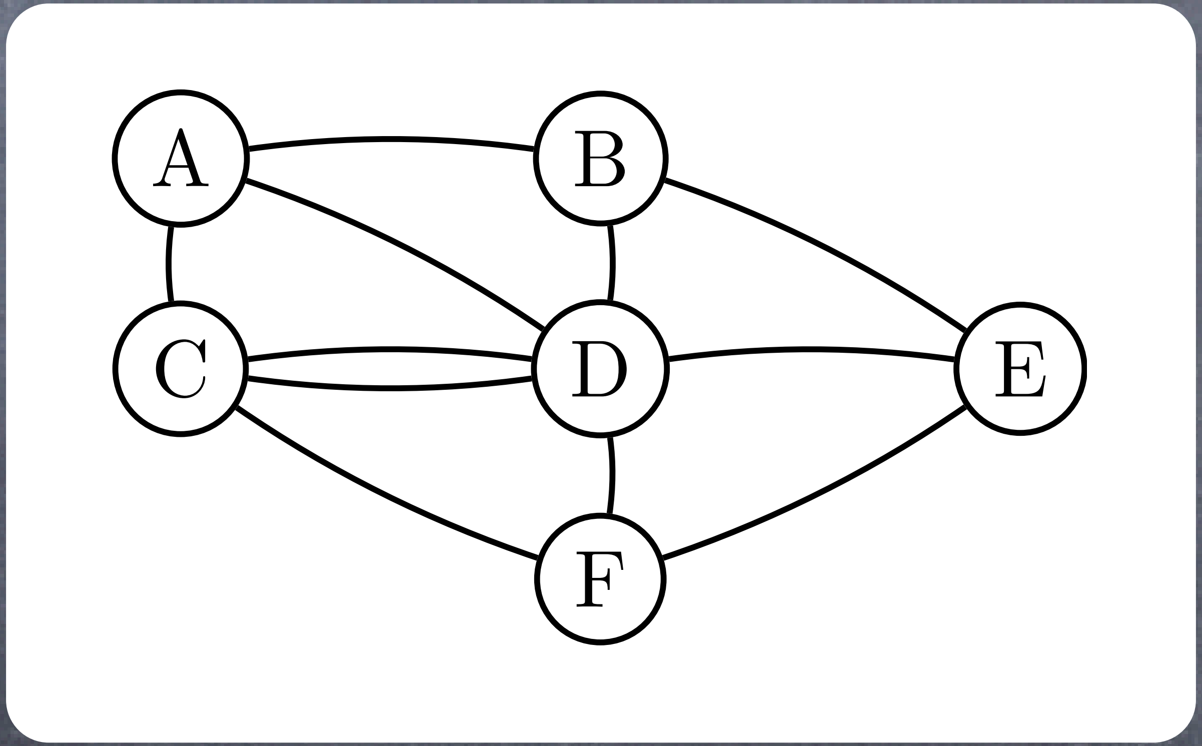
Edges with one end in V_1 and the other end in V_2 are said to cross the cut. A cut with a minimum number of edges crossing the cut is called a minimum cut.

Goal

Find a randomized algorithm to determine a minimum cut with high probability.

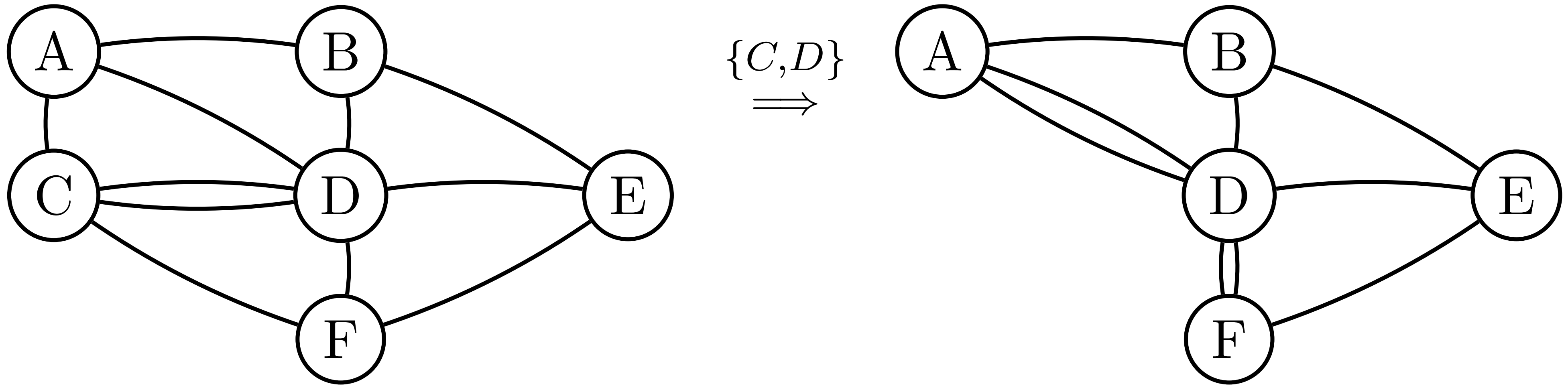
Multigraphs

A multigraph G=(V,E) is like a graph, but may contain multiple edges between vertices. Thus, E is a multiset of edges rather than a set of edges.



Edge Contraction

Given a multigraph G=(V,E) and an edge e={C,D} in E, the multigraph G/e is obtained from G by contracting the edge e, that is, by identifying the vertices C and D and removing all self-loops.



Edge Contraction

An edge in G remains in G/e with the exception of the edges e.

If e={C,D}, then any edge incident with C or D in G is incident in G/e with the merged node {C,D}.

Main Idea

A cut in G/{C,D} leads to a cut in G such that C and D are in the same block of the cut.

The size of the minimum cut of G/{C,D} is at least the size of the minimum cut of G.

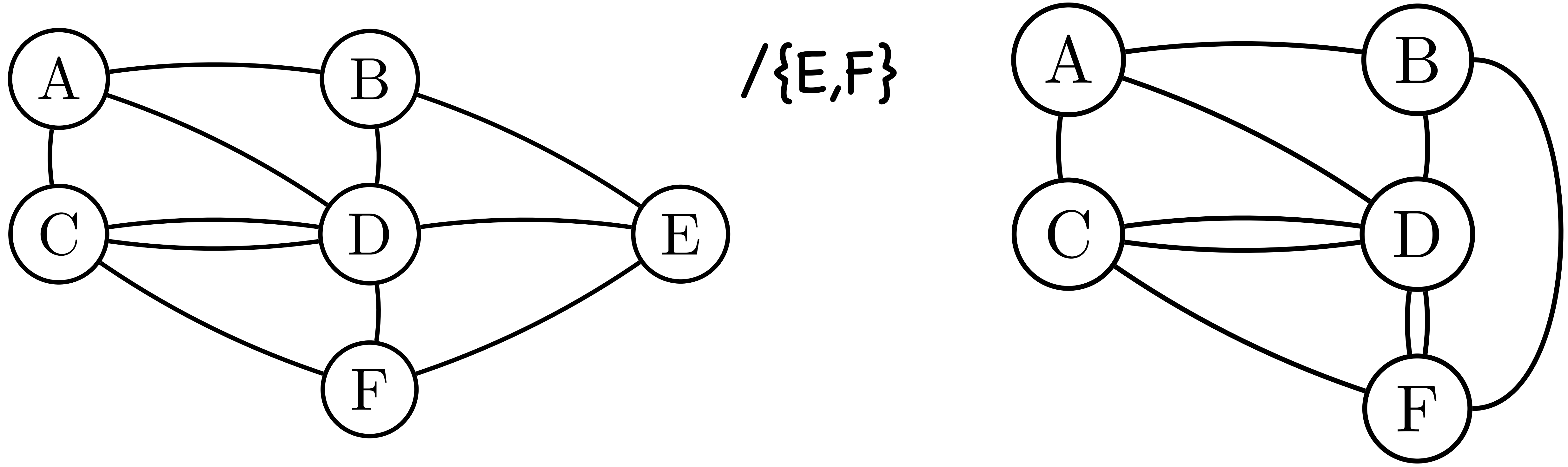
If e={C,D} did not cross a minimum cut, then G/e has the same size minimum cut than G.

If e={C,D} crosses the minimum cut, then the size of the minimum cut of G/e might be larger than the size of the minimum cut of G.

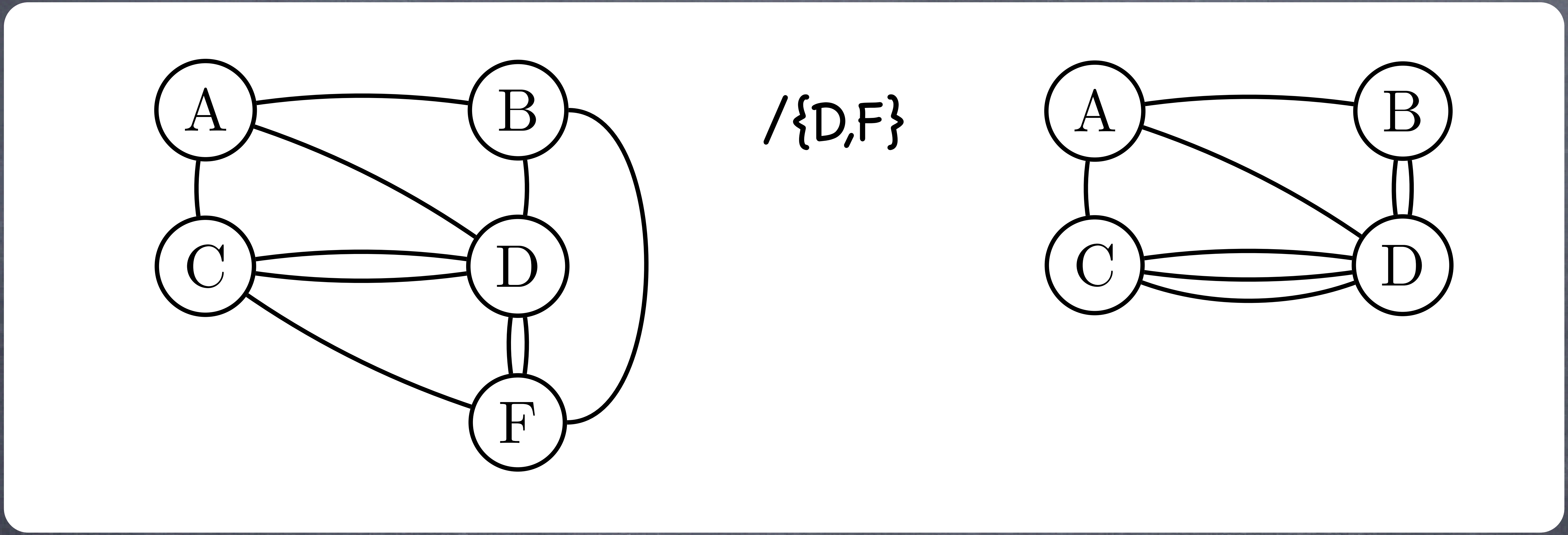
The Randomized Algorithm

```
Contract(G=(V,E)) // G is a connected loopfree multigraph with |V|>=2.

while (|V|>2) {

   Select e ∈ E uniformly at random;

   G := G/e;

}
return |E|; // |E| is an upper bound on the minimum cut of G.
```

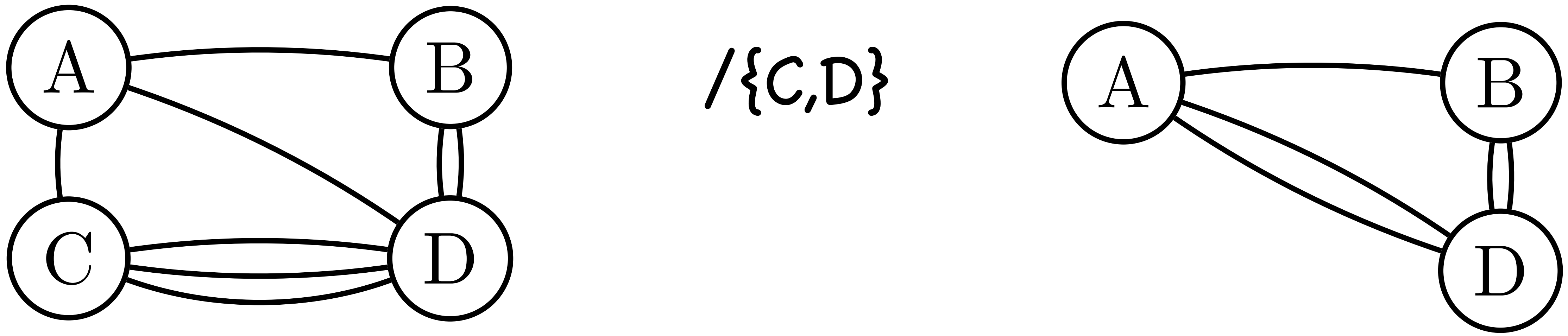
Example



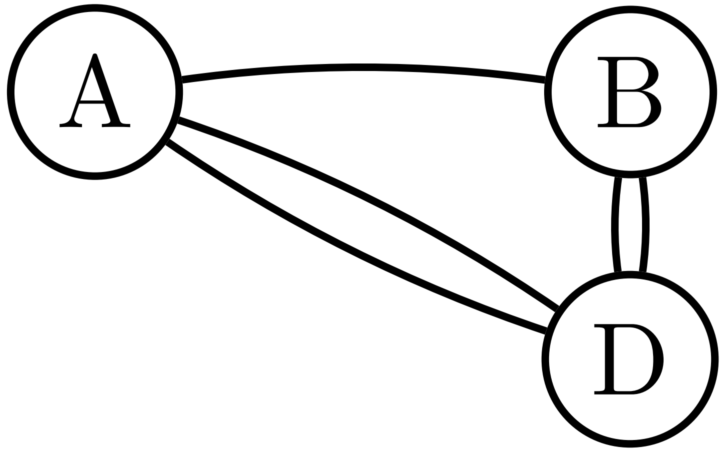
Example



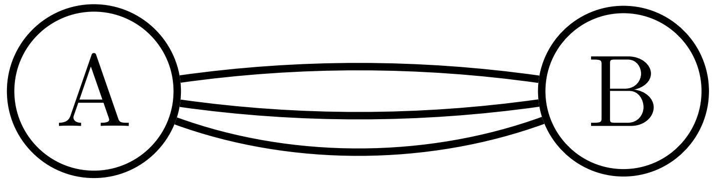
Example



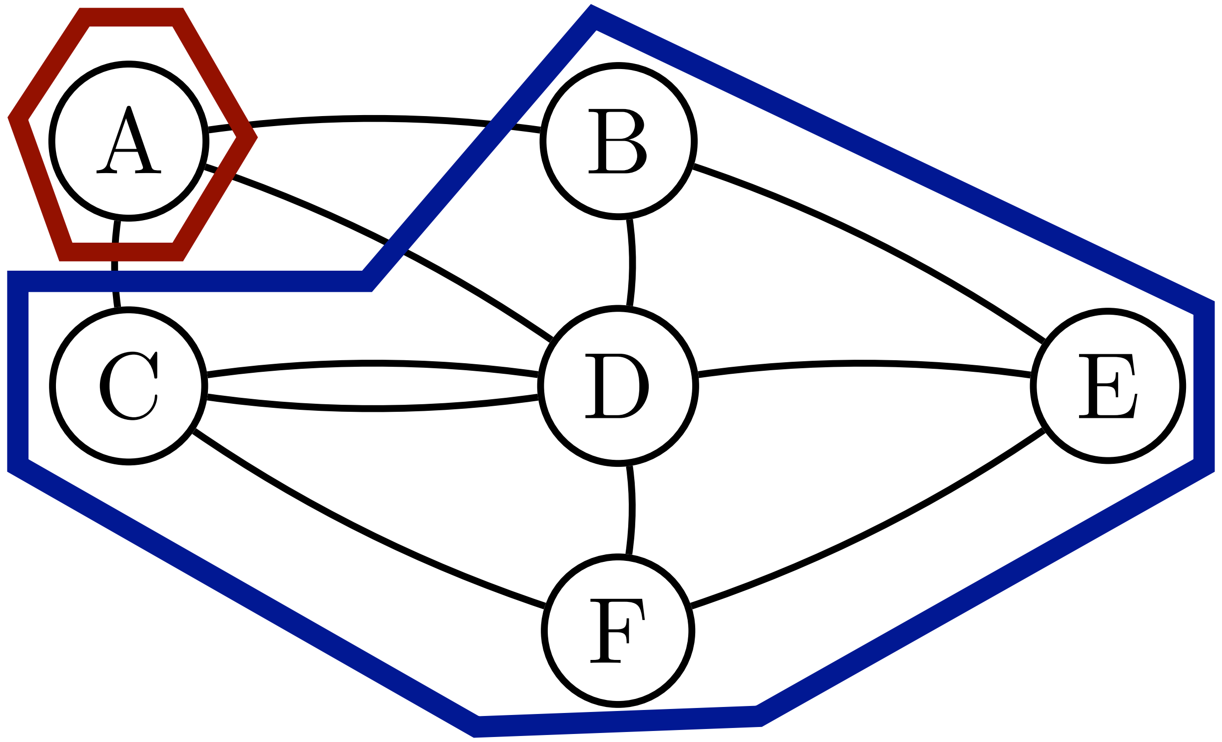
Example

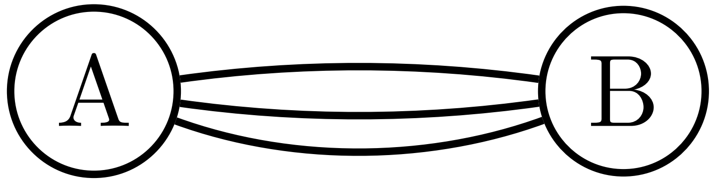


/{B,D}



Example





Contractions: {E,F}, {D,F}, {C,D}, {B,D}. Cut: {A}, {B,C,D,E,F}

Intuition

Why does it work?

If a cut is of large size, then it is likely that one of its crossing edges is selected for contraction.

If a cut is of small size, then it is less likely that one of its crossing edges is selected for contraction.

=> Algorithm has a natural bias towards minimum cuts!

Analysis

Let C be one fixed minimum cut of a multigraph G with n vertices.

Let E_k denote the event that no edge of C is picked for contraction during the k^{th} iteration of the algorithm.

Goal: Estimate $Pr[E_1 \cap E_2 \cap \ldots \cap E_{n-2}]$ = Pr[find minimum cut C]

Analysis

We have $\Pr[E \cap F] = \Pr[E|F] \Pr[F]$.

Thus, it follows that

$\Pr[E_{n-2} \cap E_{n-3} \cap \ldots \cap E_1] = \Pr[E_{n-2} | E_{n-3} \cap \ldots \cap E_1] \Pr[E_{n-3} \cap \ldots \cap E_1]$

$= \Pr[E_{n-2} | E_{n-3} \cap \ldots \cap E_1] \Pr[E_{n-3} | E_{n-4} \cap \ldots \cap E_1] \Pr[E_{n-4} \cap \ldots \cap E_1]$

$= \Pr[E_{n-2} | E_{n-3} \cap \ldots \cap E_1] \Pr[E_{n-3} | E_{n-4} \cap \ldots \cap E_1] \ldots \Pr[E_2 \cap E_1 | E_1] \Pr[E_1]$

The conditional probabilities are not difficult to calculate!

Analysis

Suppose that the size of the minimum cut is k.

This means that the degree of each vertex is at least k, hence there exist at least kn/2 edges.

The probability to select an edge crossing the cut C in the first step is at most $k/(kn/2) = 2/n$. Consequently, $Pr[E_1] \geq 1 - 2/n = (n - 2)/n$.

Analysis

As before!

At the beginning of the m^{th} step, with $m \geq 2$, there are $n - m+1$ remaining vertices. Assuming that none of the edges crossing C were selected in previous steps, the minimum cut is still at least k, hence the multigraph has at this stage at least $k(n-m+1)/2$ edges. The probability to select an edge crossing the cut C is $2/(n-m+1)$. It follows that

$$\Pr[E_m | E_{m-1} \cap \ldots \cap E_1] \geq 1 - 2/(n-m+1) = (n-m-1)/(n-m+1) .$$

Conclusion

$$\Pr\Big[\bigcap_{j=1}^{n-2} E_j\Big] \geq \prod_{m=1}^{n-2}\left(\frac{n-m-1}{n-m+1}\right) = \frac{2}{n(n-1)}.$$

Repetitions

Run the algorithm $a(n-1)n/2=a\binom{n}{2}$ times. Since $1-x <= e^{-x}$ holds for all x, the probability that one of the a runs finds the minimum cut is at least

$$1-\left(1-\frac{1}{\binom{n}{2}}\right)^{a\binom{n}{2}} \geq 1-e^{-a}$$

Choosing $a=c \ln n$, so a total of $c \ln(n) \binom{n}{2}$ repetitions yields

Pr[find minimum cut] >= $1-\exp(-c\ln n) = 1-1/n^c$.