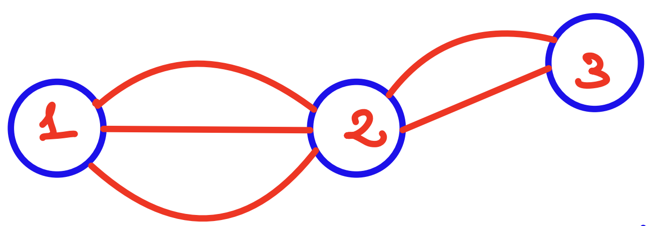
A MONTE CARLO ALGORITHM FOR MIN-CUT

OBJECTIVE: Determine the minimum cardinality of a cut in a multigraph (general)

LESSON: Show that randomization may be used to yield efficient but simple algorithms

Recall that a MULTIGRAPH $\mathcal{G}=(V,\mathcal{E})$ is defined on a finite set of nodes but the edge set $\mathcal{E}$ is a multiset (each edge has a multiplicity) (see Lecture 12)

EXAMPLE:



RECALL: The concept of path, connectivity, etc. does not change in multigraphs

DEF An edge cut in a multigraph $\mathcal{G}=(V,\mathcal{E})$ is a multisubset $\mathcal{C} \subseteq \mathcal{E}$:

$\mathcal{G}' = (V, \mathcal{E}-\mathcal{C})$ is not connected

$(\exists s,t \in V : \nexists \Pi_{st} \text{ in } \mathcal{G}')$

REMARK $\mathcal{E}-\mathcal{C}$ must take into account multiplicities. E.g. To disconnect 1 from 2 in the above example, $\mathcal{C}$ must contain $\{1,2\}$ with multiplicity $m_{12}=3$

DEF The cardinality of the cut is

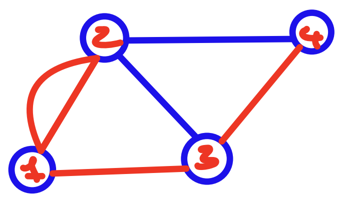
$$|\mathcal{C}| = \sum_{e \in \mathcal{C}} m_e$$

MIN-CUT PROBLEM:

determine the edge-cut $\mathcal{C}^*$ of minimum cardinality of a connected (multi)-graph $\mathcal{G}=(V,\mathcal{E})$
w.l.o.g.

OBSERVATION: A node cut $(W, V-W)$ $W \subset V$ is also an edge cut. But an edge cut is not necessarily a node cut

EXAMPLE

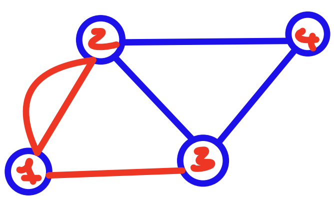


$\mathcal{C} = \{\{1,2\}, \{1,3\}, \{3,4\}\}$
$m_{12}=2$

is an edge-cut (since $\mathcal{G}' = (V, \mathcal{E}-\mathcal{C})$ is not connected)

However $\nexists\, W \subseteq V : \mathcal{C} \equiv (W, V-W)$.

IMPORTANT: An edge cut always contains a node cut!



$\mathcal{C}' = \{\{1,2\}, \{1,3\}\}$ is still an edge cut.
$m_{12}=2$

Also:

$$\mathcal{C}' = (\underbrace{\{1\}}_{W}, \underbrace{\{2,3,4\}}_{V-W})$$

EXERCISE: Determine a node cut from an edge-cut

CONSEQUENCE: A minimum edge-cut is always a node-cut!

DETERMINISTIC ALGORITHMS: (s,t)-MAX FLOW algorithm in a network with UNIT CAPACITIES provides a MINIMUM CARDINALITY NODE CUT $(W^*, V-W^*)$ with $s \in W^*$, $t \in V-W^*$ (see MAX-FLOW, MIN-CUT THEOREM)

$\Rightarrow$ must solve $|V|-1$ (s,t)-MAX FLOW instances (fixed s, vary $t \in V-\{s\}$) to obtain the MIN CUT.

RUNNING TIME: $\Omega(n^{3/2} m) = \Omega(n^{3.5...})$

ALSO: The algorithm is COMPLICATED

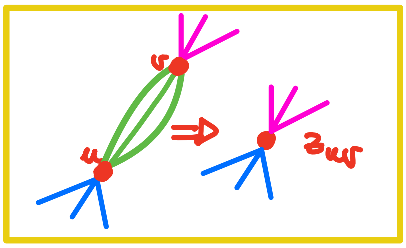
We will present very simple randomized strategies (Karger 1993, Karger-Stein 1993) for min-cut yielding a much faster algorithm (also very practical)

The algorithms are based on node contraction operations that reduce the vertex-set of G.

Contractions are performed in sequence, until the resulting graph has only two nodes z_1, z_2 ($\Rightarrow$ the cut is then $\{\{z_1, z_2\}\}$ of size $m_{\{z_1,z_2\}}$)

DEFINITION Given $\mathcal{G}=(V,\mathcal{E})$ and $e=\{u,v\}\in\mathcal{E}$ the contraction of $\mathcal{G}$ w.r.t. e is:

$\mathcal{G}/e=(V',\mathcal{E}')$ with



$V' = V-\{u,v\}\cup\{z_{u,v}\}$

$\mathcal{E}' = \mathcal{E} - \{\{\{x,y\}: (x=u)\vee(x=v)\}\} \cup$

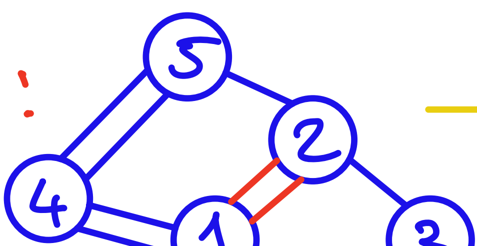
MULTISET

$\cup\{\{\{z_{u,v},y\}: (\{u,y\}\in\mathcal{E})\vee(\{v,y\}\in\mathcal{E}), y\neq u,v\}\}$

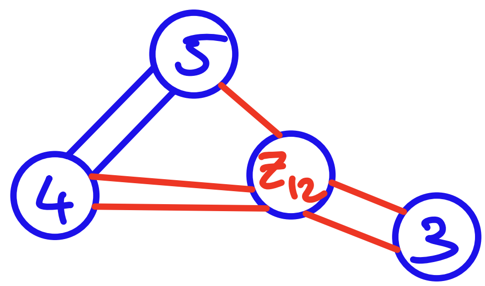
INFORMALLY: Given $e=\{u,v\}$, we fuse u and v into $z_{u,v}$, remove the m_e copies of e and substitute all other edges $\{u,y\}$ or $\{v,y\}$ with $\{z_{u,v},y\}$

EXAMPLE

$\mathcal{G}$:



$\longrightarrow$ $\mathcal{G}/_{\{1,2\}}$:



NOTE: $|V'|=|V|-1$

$|\mathcal{E}'|=|\mathcal{E}|-m(e)\leq|\mathcal{E}|-1$

The contraction operation reduces the size of the multigraph.

(CRUCIAL) PROPERTY 1 Contraction does not decrease the size of the min-cut of $\mathcal{G}/e$ w.r.t. $\mathcal{G}$

We prove the following stronger property:

PROPERTY 2: $\forall$ edge cut C' of $\mathcal{G}/e$ $\exists$ edge cut C of $\mathcal{G}$: $|C'| = |C|$

PROOF: Let $e = \{u, v\} \in \mathcal{E}$, and let $\mathcal{G}' = \mathcal{G}/e = (V', \mathcal{E}')$.

Consider a cut C' of $\mathcal{G}'$: C' disconnects $\mathcal{G}'$ into two or more connected components. Let C be the set obtained from C' by substituting each edge $\{z_{uv}, y\}$ with the "original" edge $\{u, y\}$ or $\{v, y\}$. Clearly, $|C| = |C'|$

It suffices to show that C is a cut in $\mathcal{G}$.

Consider a node $x \neq z_{uv}$ that

belongs to a different connected component from the one of z_{uv} in $(V', E' - \mathcal{C}')$. Then every path $\pi_{z_{uv}x}$ in $\mathcal{G}'$ must contain an edge $e' \in \mathcal{C}'$ (or otherwise z_{uv} and x would not be disconnected in $(V', E' - \mathcal{C}')$. It easily follows that $\mathcal{C}$ is a cut in $\mathcal{G}$, since every path from u to x in $\mathcal{G}$ must contain an edge $\hat{e} \in \mathcal{C}$ or otherwise, that path would yield a path in $\mathcal{G}'$ from z_{uv} to x not containing an edge of $\mathcal{C}'$, a contradiction.

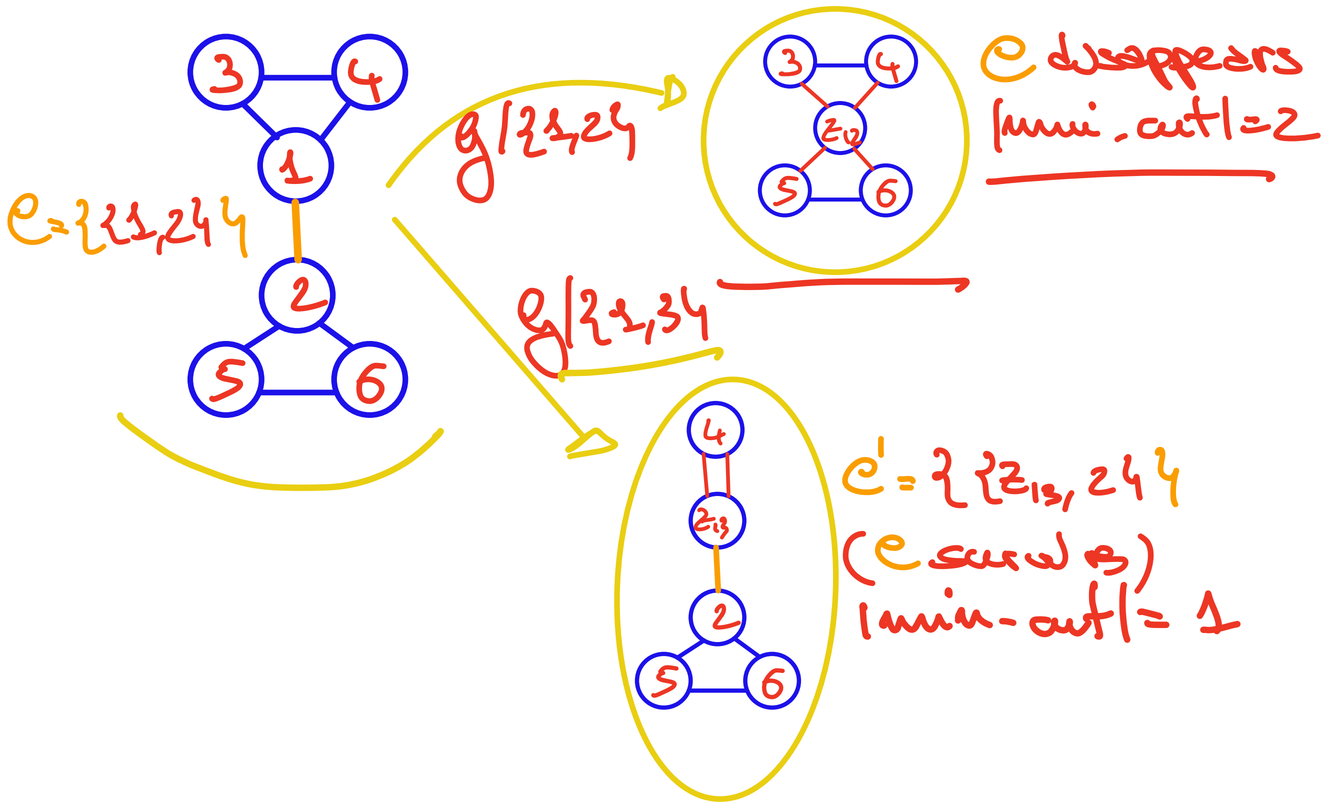
PROPERTY 1 follows as an easy COROLLARY of PROPERTY 2: indeed,

$$\{|\mathcal{C}'| : \mathcal{C}' \text{ cut of } \mathcal{G}/e\} \subseteq \{|\mathcal{C}| : \mathcal{C} \text{ cut of } \mathcal{G}\}$$

$$\Rightarrow \min\{|\mathcal{C}'| : \mathcal{C}' \text{ cut of } \mathcal{G}/e\} \geq \min\{|\mathcal{C}| : \mathcal{C} \text{ cut of } \mathcal{G}\}$$

Alternative interpretation of PROPERTIES 1 and 2: If a cut $\mathcal{C}$ in $\mathcal{G}$ does not contain e, then the cut survives in G/e, otherwise the cut disappears.

EXAMPLE:

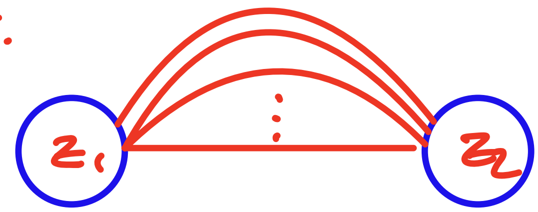


EXERCISE: Prove formally that if C is a cut in G and $e \notin C$, then the corresponding edge set C' in G/e is a cut (modify the proof of PROPERTY 2)

MORALE: Consider a fixed min-cut C^* of G. If I apply a sequence of contractions to G w.r.t. $e_1, e_2, \dots$: $G_0 = G$; $G_i = G_{i-1}/e_i$, if $\boxed{e_1, e_2, \dots \notin C^*}$, then the multisets C^*_i corresponding to C^* in G_i are still a cut of G_i

Also, since the size of the min-cut after a contraction cannot decrease, e_i^{*} is a min-cut of $\mathcal{G}_i$.

IDEA: If I perform $|V|-2$ contractions, I reduce $\mathcal{G}$ to a multigraph with only two nodes:



If the $|V|-2$ contractions avoid the edges of a fixed min-cut, then e^* corresponds to $\{\{z_1, z_2\}\}$!

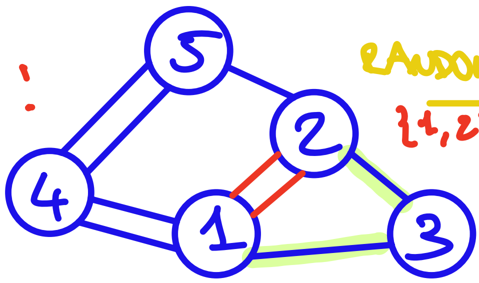
IDEA BEHIND THE RANDOMIZED ALGORITHM: perform contractions w.r.t. random edges.

SUBROUTINE:

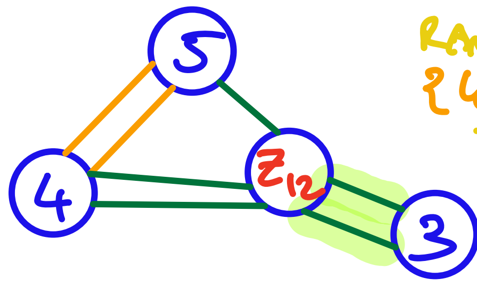
FULL_CONTRACTION $(\mathcal{G} = (V, \mathcal{E}))$
  * Let $\mathcal{G}' = (V', \mathcal{E}') = \mathcal{G}$ *
  for $i \leftarrow 1$ to $|V|-2$ do
    $e \leftarrow$ RANDOM $(\mathcal{E}')$
    {account for multiplicities}
    $\mathcal{G}' \leftarrow \mathcal{G}'/e$
  $\{|V'| = 2\}$
  return $|\mathcal{E}'|$

EXAMPLE

$\mathcal{G}$:

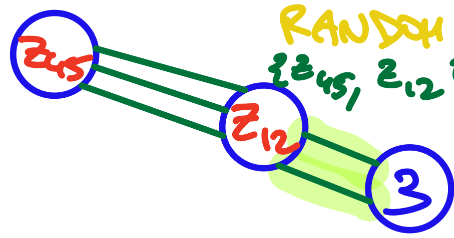


RANDOM: $\{1,2\}$ → $\mathcal{G}/\{1,2\}$:



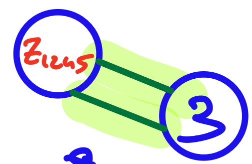
RANDOM: $\{4,5\}$ →

→ $(\mathcal{G}/\{1,2\})/\{4,5\}$:



RANDOM: $\{z_{45}, z_{12}\}$

→ $((\mathcal{G}/\{1,2\})/\{4,5\})/\{z_{45}, z_{12}\}$:



The min-cut survives!

RUNNING TIME

It is easy to see that FULL_CONTRACTION can be implemented in $O(|V|^2)$ time
(1. random edge selection in $O(|V|)$ time
2. remove occurrences of contracted edge $e=\{u,v\}$ and merge adjacences of u and v to create the adjacency of z_{uv}) ← EXERCISE

The Karger algorithm repeats FULL-CONTRACTION s times, (s to be set by the analysis)

```
KARGER(G, s)
  min ← +∞
  repeat s times
    t ← FULL-CONTRACTION(G)
    if (t < min) then min ← t
  return min {return smallest cut}
```

(We only return the size of the cut. Returning the edges is an easy extension EXERCISE)

IDEA OF THE ANALYSIS If the random edge selections in FULL-CONTRACTION miss the edges of C^*, then C^* survives and is returned. We will show that this happens with small but non negligible probability!

Then, we repeat FULL-CONTRACTION s times (as we did for MR(n,s)) to AMPLIFY THIS PROBABILITY!

ANALYSIS

We need to recall some elementary probability:

DEF E_1, E_2 are independent if:

$$\Pr(E_1 \cap E_2) = \Pr(E_1) \cdot \Pr(E_2)$$

DEF Conditional probability. If $E_1 \neq \emptyset$:

$$\Pr(E_2 | E_1) = \frac{\Pr(E_1 \cap E_2)}{\Pr(E_1)}$$

Observe that:

$$\Pr(E_1 \cap E_2) = \Pr(E_1) \cdot \Pr(E_2 | E_1)$$

This can be extended to k events:

$$\Pr(E_1 \cap E_2 \cap \dots \cap E_k) = \Pr(E_1) \cdot \Pr(E_2 | E_1) \times$$
$$\times \Pr(E_3 | E_1 \cap E_2) \cdot \ldots \cdot \Pr(E_i | E_1 \cap \dots \cap E_{i-1}) \cdot \ldots \times$$
$$\times \Pr(E_k | E_1 \cap E_2 \cap \dots \cap E_{k-1})$$
$$= \Pr(E_1) \cdot \prod_{i=2}^{k} \Pr\Big(E_i \Big| \bigcap_{j=1}^{i-1} E_j\Big)$$

This can be easily proved by induction on $k \geq 2$:

HOME

B $(k=2)$ $\Pr(E_1 \cap E_2) = \Pr(E_1) \Pr(E_2 | E_1)$ ✓
(from definition of cond. prob.)

HP → TH $\Pr(\underbrace{E_1 \cap \dots \cap E_{k-1}} \cap E_k) =$

$= \Pr(E_1 \cap \dots \cap E_{k-1}) \cdot \Pr(E_k | E_1 \cap \dots \cap E_{k-1})$ (def)

$= \overbrace{\Pr(E_1) \cdot \Pr(E_2 | E_1) \cdot \ldots \cdot \Pr(E_{k-1} | E_1 \cap \dots \cap E_{k-2})}^{HP} \cdot \Pr(E_k | E_1 \cap \dots \cap E_{k-1})$

Conditional probabilities are important in the analysis of randomized algorithms since they allow to evaluate the probability of a sequence of choices performed by the algorithm at different times.