A New Contraction Technique with Applications to Congruency-Constrained Cuts

Martin Nägele Rico Zenklusen

ETH Zürich

Introduction: Congruency-Constrained Cuts

Problem Setting, Motivation, and Our Results

Problem Setting

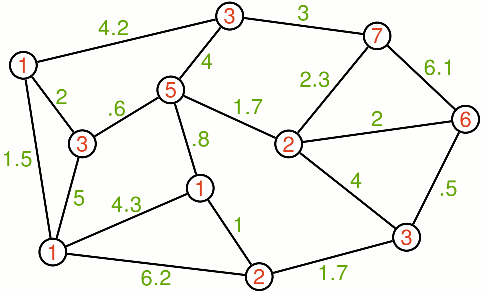
Congruency-Constrained Minimum Cut Problem (CCMC)

Input: Graph $G = (V, E)$, edge weights $w\colon E \to \mathbb{R}_{\geqslant 0}$, vertex multiplicities $\gamma\colon V \to \mathbb{Z}_{\geqslant 0}$, $m \in \mathbb{Z}_{>0}$, and $r \in \mathbb{Z}_{\geqslant 0}$.

Goal: Find a minimizer of $\min \left\{ w(\delta(C)) \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ \sum_{v \in C} \gamma(v) \equiv r \pmod{m} \end{array} \right\}$.

$m = 5$

$r = 2$



Problem Setting

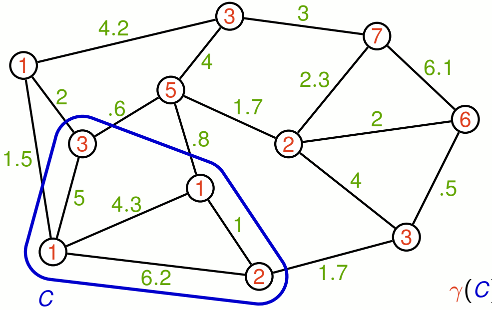
Congruency-Constrained Minimum Cut Problem (CCMC)

Input: Graph $G = (V, E)$, edge weights $w\colon E \to \mathbb{R}_{\geqslant 0}$, vertex multiplicities $\gamma\colon V \to \mathbb{Z}_{\geqslant 0}$, $m \in \mathbb{Z}_{>0}$, and $r \in \mathbb{Z}_{\geqslant 0}$.

Goal: Find a minimizer of $\min \left\{ w(\delta(C)) \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ \sum_{v\in C} \gamma(v) \equiv r \pmod{m} \end{array} \right\}$.

$m = 5$

$r = 2$



$w(\delta(C)) = 6.6$

$\gamma(C) \equiv 2 \pmod{5}$

Motivation and Prior Results

- Generalization of well-known cut problems:
  - ⇝ Global minimum cuts, minimum s-t-cuts, minimum odd cuts.
- Integer Programming with bounded subdeterminants:

  Can $\min\{c^\top x \mid Ax \leqslant b,\ x \in \mathbb{Z}^n\}$ be solved efficiently if $A \in \mathbb{Z}^{m \times n}$ is m-modular?

  - ⇝ *Bimodular integer programming* ($m = 2$):
    Reduction to parity-constrained cut and flow problems. [Artmann, Weismantel, Zenklusen, 2017]
  - ⇝ CCMC can be reduced to m-modular ILPs.
- Congruency-constrained submodular minimization:
  - ⇝ Efficient algorithm for prime power moduli. [Nägele, Sudakov, Zenklusen, 2018]
  - ⇝ Barriers for composite moduli. [Gopi, 2019]

Our Results

Theorem 1: PRAS for CCMC

CCMC with constant modulus m admits a polynomial time randomized approximation scheme.

Our Results

Theorem 1: PRAS for CCMC

CCMC with constant modulus m admits a polynomial time randomized approximation scheme.

- Approach inspired by Karger's contraction algorithm.
- Novel way of sampling vertex pairs to contract.
  - $\rightsquigarrow$ Using splitting-off techniques from Graph Theory.
- Combination with approximate reduction steps.

Our Results

Theorem 1: PRAS for CCMC

CCMC with constant modulus m admits a polynomial time randomized approximation scheme.

Theorem 2: Exact algorithm for special case

CCMC with modulus $m = pq$ for primes $p \neq q$ admits an exact polynomial time randomized algorithm.

Our Results

Theorem 1: PRAS for CCMC

CCMC with constant modulus m admits a polynomial time randomized approximation scheme.

Theorem 2: Exact algorithm for special case

CCMC with modulus $m = pq$ for primes $p \neq q$ admits an exact polynomial time randomized algorithm.

Theorem 3: Structure for instances with prime moduli

Given a CCMC problem with prime modulus and nonzero optimal value denoted by OPT, there is a randomized algorithm returning polynomially many s-t cut problems such that w.h.p.,

C is solution of (CCMC) problem with value $\leqslant \kappa \cdot \mathrm{OPT}$ $\iff$ C is solution of one of the s-t cut problems with value $\leqslant \kappa \cdot \mathrm{OPT}$.

Karger’s Contraction Algorithm...

. . . and how to adopt it for CCMC.

Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***

Contract a random edge.

return Cut corresponding to a remaining vertex.

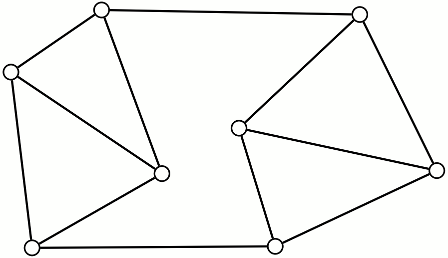
Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***

Contract a random edge.

return Cut corresponding to a remaining vertex.



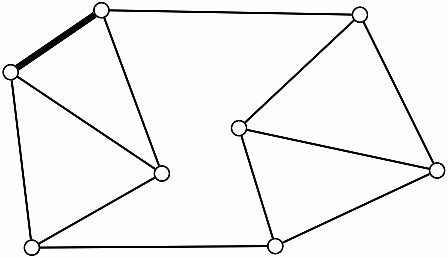
Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***

Contract a random edge.

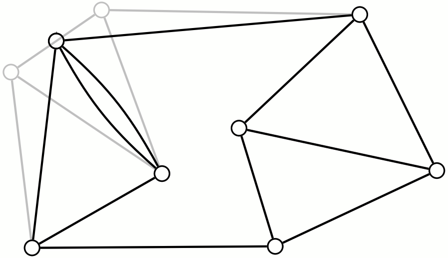
return Cut corresponding to a remaining vertex.



Karger's Contraction Algorithm

Algorithm

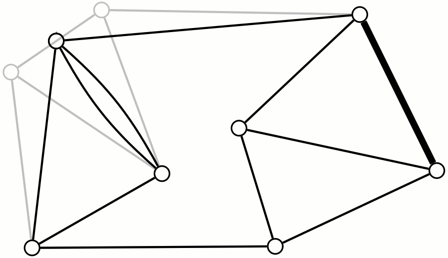
while $|V| > 2$ ***do:***
  Contract a random edge.
return Cut corresponding to a remaining vertex.



Karger's Contraction Algorithm

Algorithm

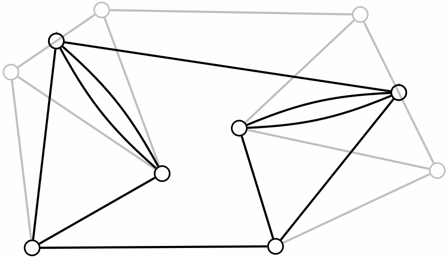
while $|V| > 2$ ***do:***
Contract a random edge.
return Cut corresponding to a remaining vertex.



Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***
  Contract a random edge.
return Cut corresponding to a remaining vertex.



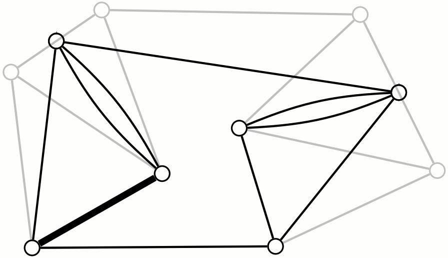
Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***

Contract a random edge.

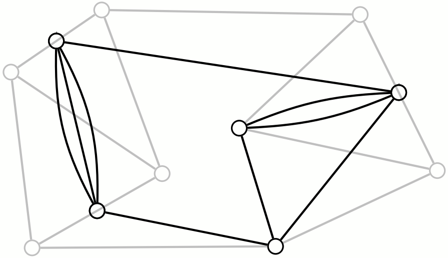
return Cut corresponding to a remaining vertex.



Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***
Contract a random edge.
return Cut corresponding to a remaining vertex.



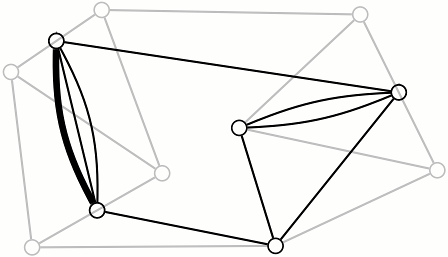
Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***

Contract a random edge.

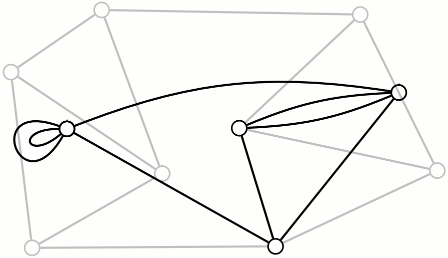
return Cut corresponding to a remaining vertex.



Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***
  Contract a random edge.
return Cut corresponding to a remaining vertex.



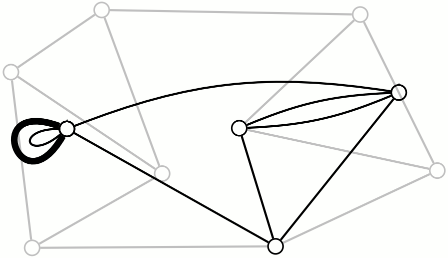
Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***

Contract a random edge.

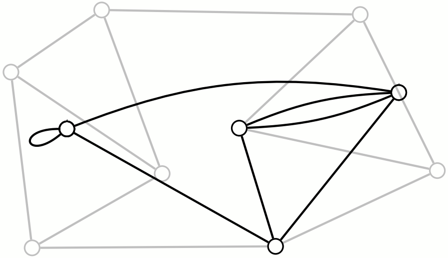
return Cut corresponding to a remaining vertex.



Karger's Contraction Algorithm

Algorithm

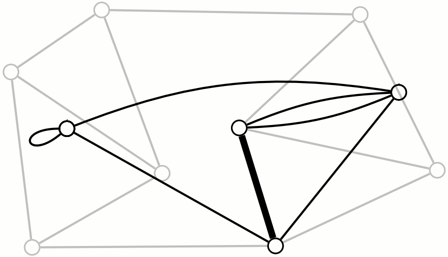
while $|V| > 2$ ***do:***
  Contract a random edge.
return Cut corresponding to a remaining vertex.



Karger's Contraction Algorithm

Algorithm

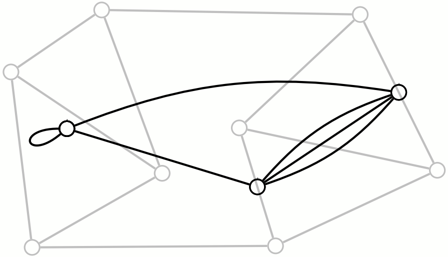
while $|V| > 2$ ***do:***
  Contract a random edge.
return Cut corresponding to a remaining vertex.



Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***
  Contract a random edge.
return Cut corresponding to a remaining vertex.



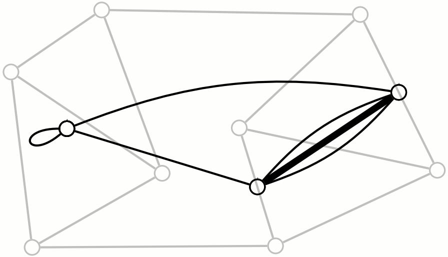
Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***

  Contract a random edge.

return Cut corresponding to a remaining vertex.



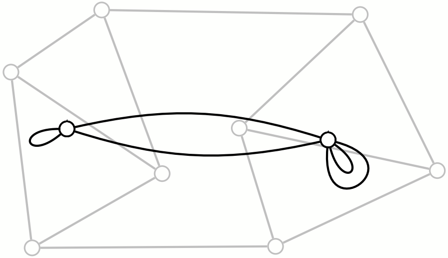
Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***

Contract a random edge.

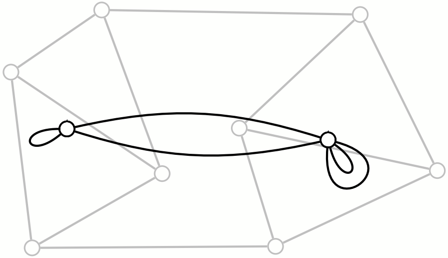
return Cut corresponding to a remaining vertex.



Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***
  Contract a random edge.
return Cut corresponding to a remaining vertex.



Analysis:

- Singletons are feasible solution candidates.

$$\implies |\delta(v)| \geqslant \mathrm{OPT}\ .$$

- Contractions uniformly at random:

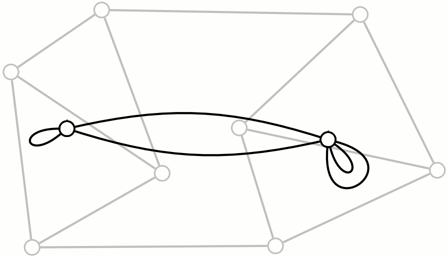
$$\Pr\left[\begin{matrix}\text{contraction is}\\ \text{bad wrt. } C_{\mathrm{OPT}}\end{matrix}\right] = \frac{\mathrm{OPT}}{|E|} = \frac{\mathrm{OPT}}{\frac{1}{2}\sum_{v\in V}|\delta(v)|} \leqslant \frac{2}{|V|}$$

$$\implies \Pr\left[\begin{matrix}\text{no bad}\\ \text{contraction}\end{matrix}\right] \geqslant \prod_{i=3}^{|V|}\left(1-\frac{2}{i}\right) = \Omega\left(\frac{1}{|V|^2}\right).$$

Karger's Contraction Algorithm

Algorithm

while $|V| > 2$ ***do:***
  Contract a random edge.
return Cut corresponding to a remaining vertex.



Analysis:

- Singletons are feasible solution candidates.

$$\implies |\delta(v)| \geqslant \mathrm{OPT}\ .$$

- Contractions uniformly at random:

$$\Pr\left[\begin{array}{c}\text{contraction is}\\ \text{bad wrt. } C_{\mathrm{OPT}}\end{array}\right] = \frac{\mathrm{OPT}}{|E|} = \frac{\mathrm{OPT}}{\frac{1}{2}\sum_{v\in V}|\delta(v)|} \leqslant \frac{2}{|V|}$$

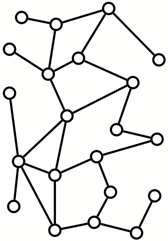
$$\implies \Pr\left[\begin{array}{c}\text{no bad}\\ \text{contraction}\end{array}\right] \geqslant \prod_{i=3}^{|V|}\left(1-\frac{2}{i}\right) = \Omega\left(\frac{1}{|V|^2}\right).$$

If $\sum_{v\in V}|\delta(v)| \geqslant \varepsilon\cdot|V|\cdot\mathrm{OPT}$, Karger until ${}^2/_\varepsilon$ vertices remain succeeds with probability $\Omega(|V|^{-{}^2/_\varepsilon})$.

⇝ Enumerate remaining options.

What fails with $\gamma(C) \equiv r \pmod{m}$?

$G = (V, E)$

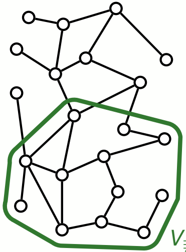


Problems:

- Singletons are generally not feasible.
- Average degree can be small.
- Edge contractions might not be enough.

What fails with $\gamma(C) \equiv r \pmod{m}$?

$G = (V, E)$

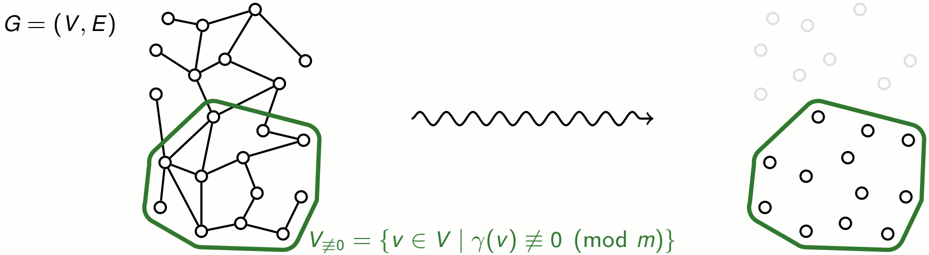


$V_{\not\equiv 0} = \{v \in V \mid \gamma(v) \not\equiv 0 \pmod{m}\}$

Problems:

- Singletons are generally not feasible.
- Average degree can be small.
- Edge contractions might not be enough.

What fails with $\gamma(C) \equiv r \pmod{m}$?



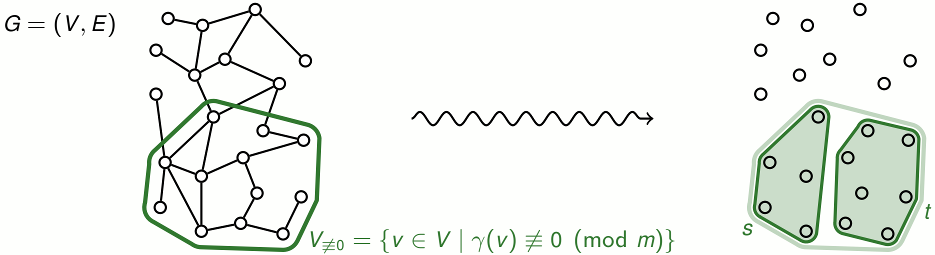
Problems:

- Singletons are generally not feasible.
- Average degree can be small.
- Edge contractions might not be enough.

Plan:

- Reduce to $V_{\not\equiv 0}$, allow contracting arbitrary vertex pairs.
- Extend to $V \setminus V_{\not\equiv 0}$ solving unconstrained s-t cut problem.

What fails with $\gamma(C) \equiv r \pmod{m}$?



Problems:

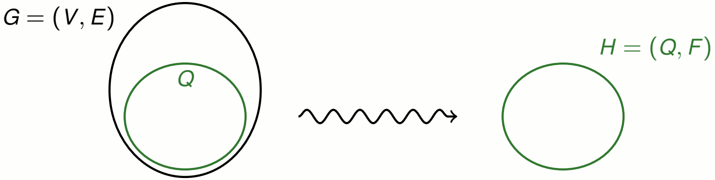
- Singletons are generally not feasible.
- Average degree can be small.
- Edge contractions might not be enough.

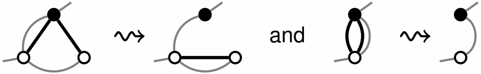
Plan:

- Reduce to $V_{\not\equiv 0}$, allow contracting arbitrary vertex pairs.
- Extend to $V \setminus V_{\not\equiv 0}$ solving unconstrained s-t cut problem.

Splitting-Off: Auxiliary Graph Construction

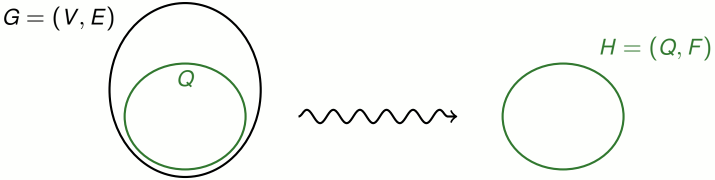
- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]

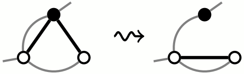
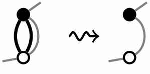


- Two operations: 

Splitting-Off: Auxiliary Graph Construction

- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



- Two operations:  and 

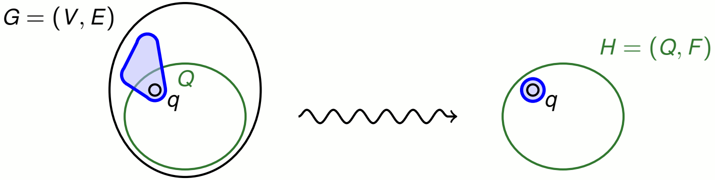
Theorem [Lov. '76]

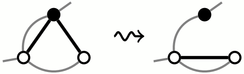
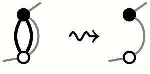
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

Splitting-Off: Auxiliary Graph Construction

- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



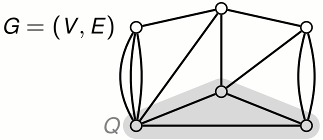
- Two operations:  and 

Theorem [Lov. '76]

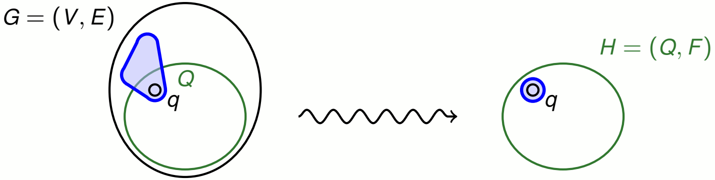
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

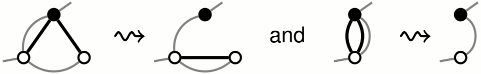
- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

Splitting-Off: Auxiliary Graph Construction



- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



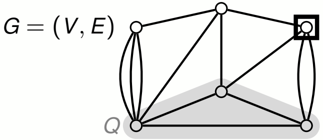
- Two operations: 

Theorem [Lov. '76]

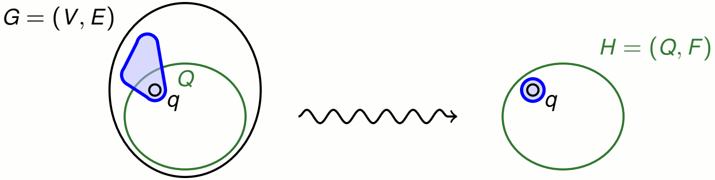
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{l} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

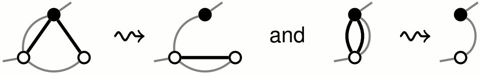
Splitting-Off: Auxiliary Graph Construction



- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



- Two operations:

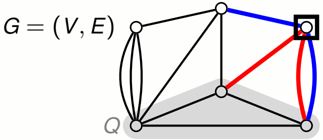


Theorem [Lov. '76]

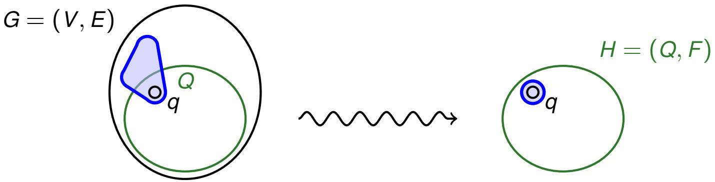
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

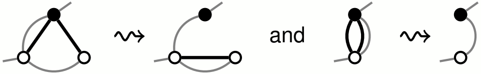
- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

Splitting-Off: Auxiliary Graph Construction



- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



- Two operations: 

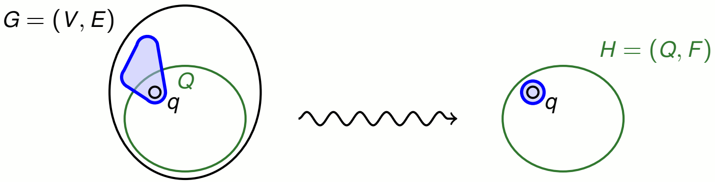
Theorem [Lov. '76]

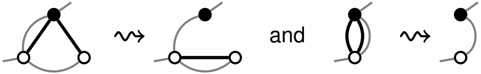
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

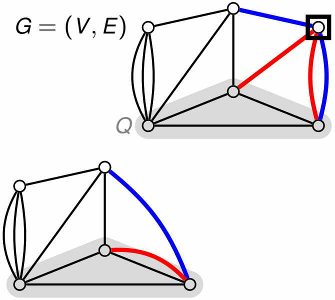
- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

Splitting-Off: Auxiliary Graph Construction

- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



- Two operations: 



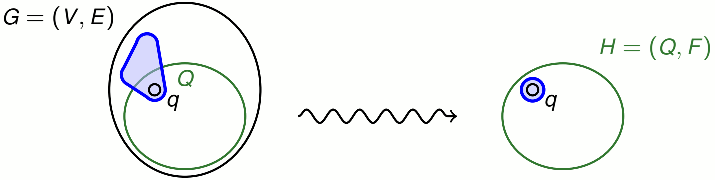
Theorem [Lov. '76]

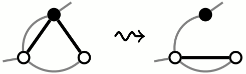
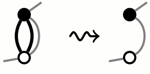
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

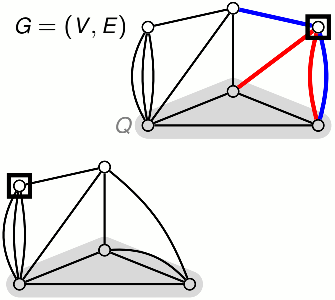
- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

Splitting-Off: Auxiliary Graph Construction

- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



- Two operations:  and 



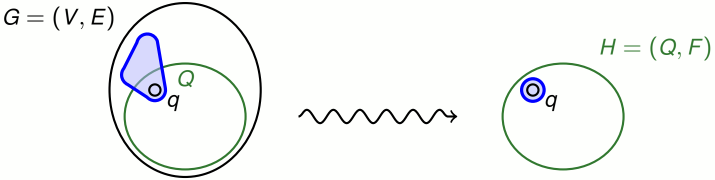
Theorem [Lov. '76]

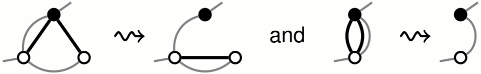
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

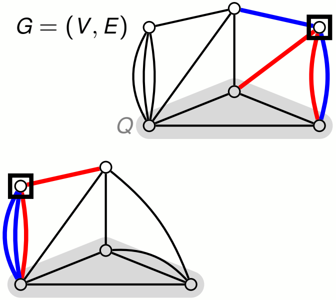
- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{l} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

Splitting-Off: Auxiliary Graph Construction

- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



- Two operations: 



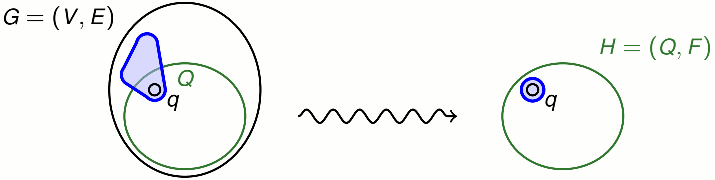
Theorem [Lov. '76]

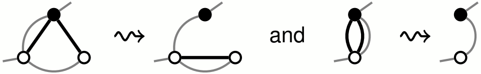
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

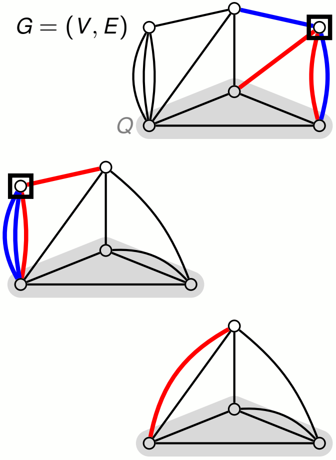
- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

Splitting-Off: Auxiliary Graph Construction

- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



- Two operations: 



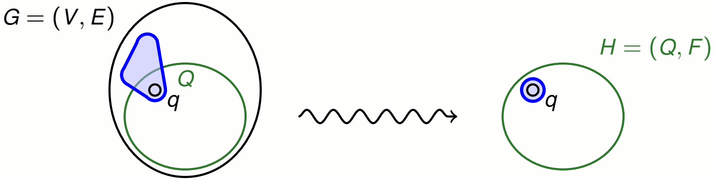
Theorem [Lov. '76]

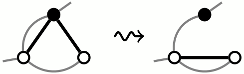
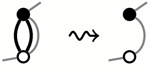
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

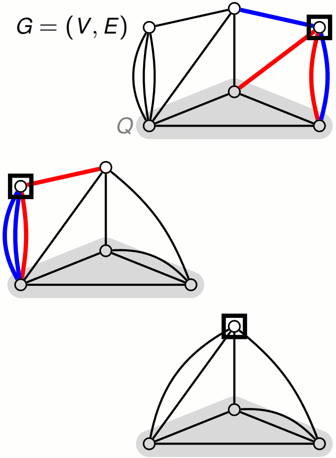
- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

Splitting-Off: Auxiliary Graph Construction

- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



- Two operations:  and 



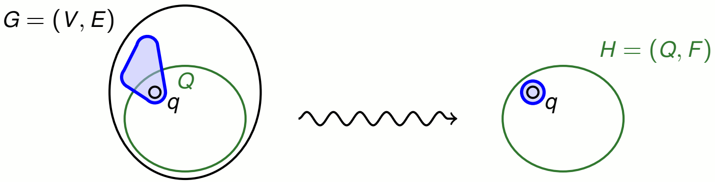
Theorem [Lov. '76]

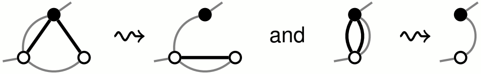
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

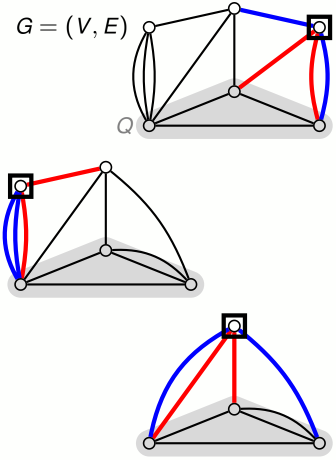
- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

Splitting-Off: Auxiliary Graph Construction

- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



- Two operations: 



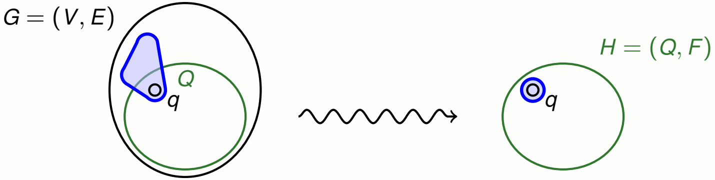
Theorem [Lov. '76]

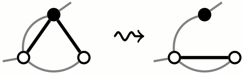
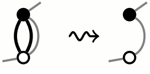
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

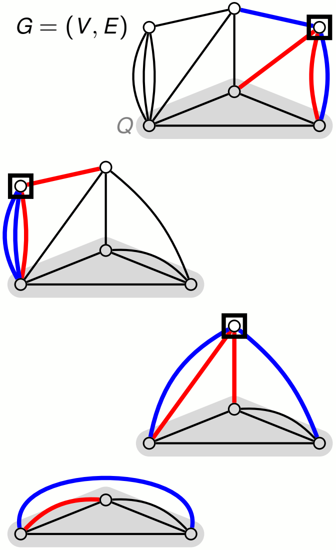
- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

Splitting-Off: Auxiliary Graph Construction

- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



- Two operations:  and 



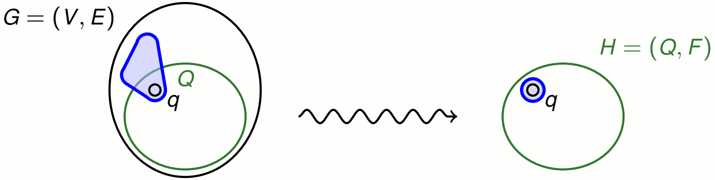
Theorem [Lov. '76]

Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

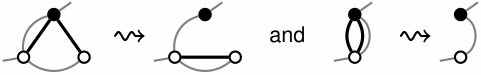
- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

Splitting-Off: Auxiliary Graph Construction

- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



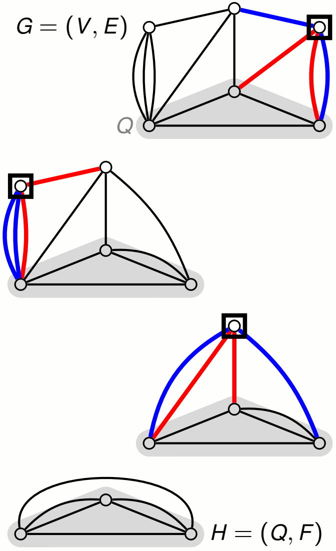
- Two operations:



Theorem [Lov. '76]

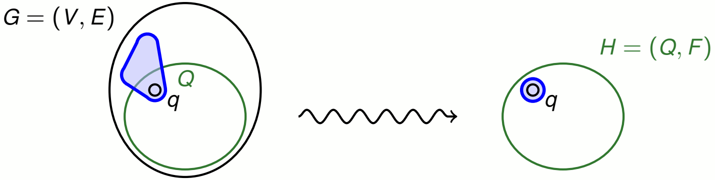
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

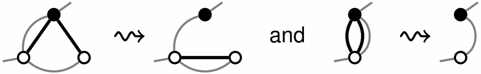
- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{c} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

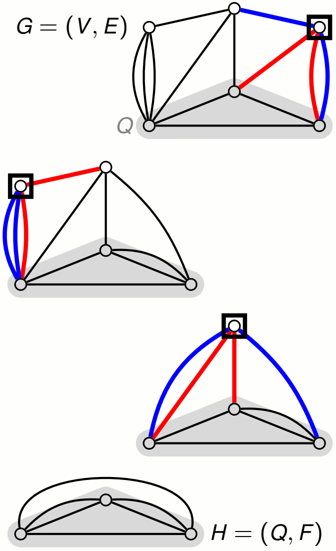


Splitting-Off: Auxiliary Graph Construction

- Fundamental technique from Graph Theory [Lovász, 1976 & 1979] [Mader, 1978]



- Two operations: 



Theorem [Lov. '76]

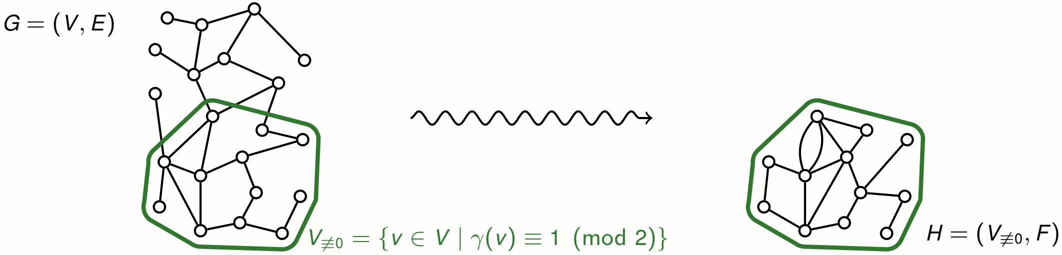
Let G Eulerian, then edges can be split from $v \in V \setminus Q$ in pairs such that

- cut values do not increase, and
- $\nu(\{q\}) := \min \left\{ |\delta_G(C)| \;\middle|\; \begin{array}{l} \emptyset \subsetneq C \subsetneq V, \\ C \cap Q = \{q\} \end{array} \right\}$ is preserved for all $q \in Q$.

- Weighted algorithmic version: Combination with ideas of Frank. [Frank, 1992]

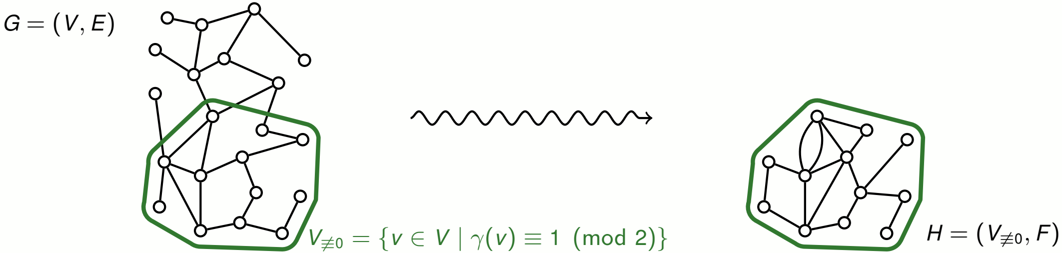
A Contraction-Approach to Odd Cuts

- CCMC with $m = 2$ and $r = 1$, i.e., constraint $\gamma(C) \equiv 1 \pmod 2$.



A Contraction-Approach to Odd Cuts

- CCMC with $m = 2$ and $r = 1$, i.e., constraint $\gamma(C) \equiv 1 \pmod 2$.



- Optimal cut value did not increase.

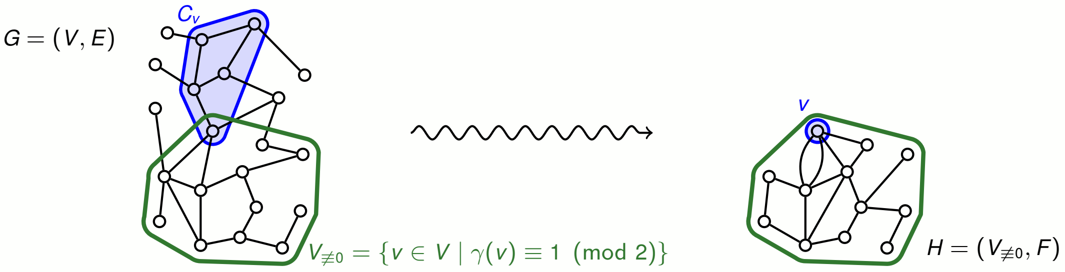
  $\rightsquigarrow$ $|\delta_H(C_{\mathrm{OPT}} \cap V_{\not\equiv 0})| \leqslant |\delta_H(C_{\mathrm{OPT}})| = \mathrm{OPT}$.

- Singletons in H correspond to feasible solutions.

  $\rightsquigarrow$ $|\delta_H(v)| = |\delta_G(C_v)| \geqslant \mathrm{OPT}$.

A Contraction-Approach to Odd Cuts

- CCMC with $m = 2$ and $r = 1$, i.e., constraint $\gamma(C) \equiv 1 \pmod 2$.



- Optimal cut value did not increase.

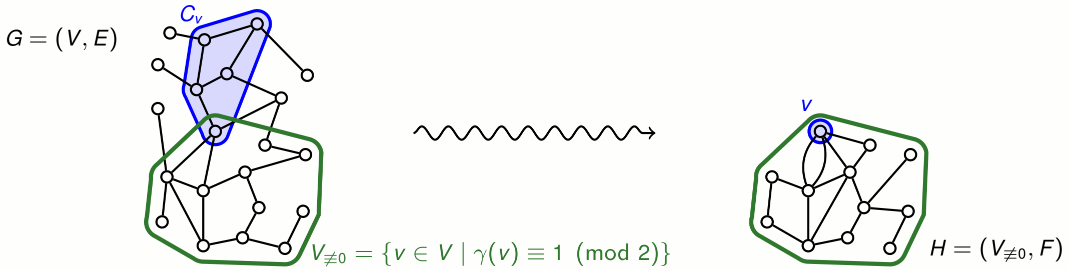
  ⇝ $|\delta_H(C_{\mathrm{OPT}} \cap V_{\not\equiv 0})| \leqslant |\delta_H(C_{\mathrm{OPT}})| = \mathrm{OPT}$.

- Singletons in H correspond to feasible solutions.

  ⇝ $|\delta_H(v)| = |\delta_G(C_v)| \geqslant \mathrm{OPT}$.

A Contraction-Approach to Odd Cuts

- CCMC with $m = 2$ and $r = 1$, i.e., constraint $\gamma(C) \equiv 1 \pmod 2$.



- Optimal cut value did not increase.

  $\rightsquigarrow$ $|\delta_H(C_{\mathrm{OPT}} \cap V_{\not\equiv 0})| \leqslant |\delta_H(C_{\mathrm{OPT}})| = \mathrm{OPT}$.

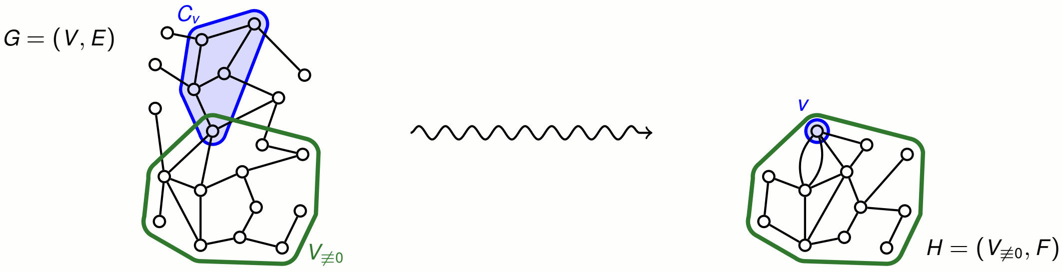
- Singletons in H correspond to feasible solutions.

  $\rightsquigarrow$ $|\delta_H(v)| = |\delta_G(C_v)| \geqslant \mathrm{OPT}$.

$\Longrightarrow$ Karger-type analysis with respect to $V_{\not\equiv 0}$ works!

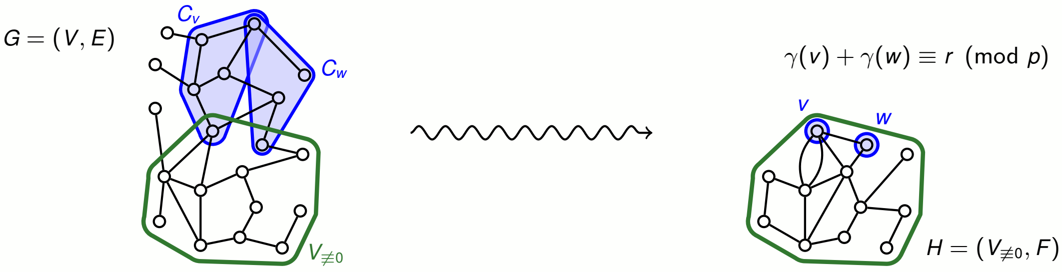
CCMC with Prime Modulus p

- **Issue:** Singletons in H do not necessarily correspond to cuts with $\gamma(C) \equiv r \pmod{p}$.



CCMC with Prime Modulus p

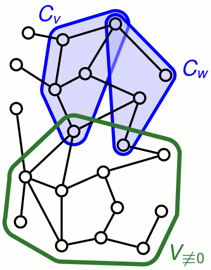
- **Issue:** Singletons in H do not necessarily correspond to cuts with $\gamma(C) \equiv r \pmod{p}$.



CCMC with Prime Modulus p

- **Issue:** Singletons in H do not necessarily correspond to cuts with $\gamma(C) \equiv r \pmod{p}$.

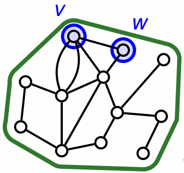
$G = (V, E)$



$$|\delta_H(v)| + |\delta_H(w)| = |\delta_G(C_v)| + |\delta_G(C_w)|$$
$$\geqslant |\delta_G(C_v \cup C_w)| \geqslant \mathrm{OPT}$$

$\rightsquigarrow$

$$\gamma(v) + \gamma(w) \equiv r \pmod{p}$$

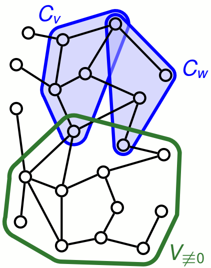


$H = (V_{\neq 0}, F)$

CCMC with Prime Modulus p

- **Issue:** Singletons in H do not necessarily correspond to cuts with $\gamma(C) \equiv r \pmod p$.

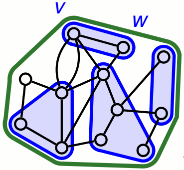
$G = (V, E)$



$$|\delta_H(v)| + |\delta_H(w)| = |\delta_G(C_v)| + |\delta_G(C_w)|$$
$$\geqslant |\delta_G(C_v \cup C_w)| \geqslant \mathrm{OPT}$$

$$\rightsquigarrow$$

$$\gamma(v) + \gamma(w) \equiv r \pmod p$$



$H = (V_{\not\equiv 0}, F)$

Thm. **(Cauchy-Davenport)** Among any p nonzero elements of $\mathbb{Z}/p\mathbb{Z}$, there is a subset summing to $r \pmod p$.

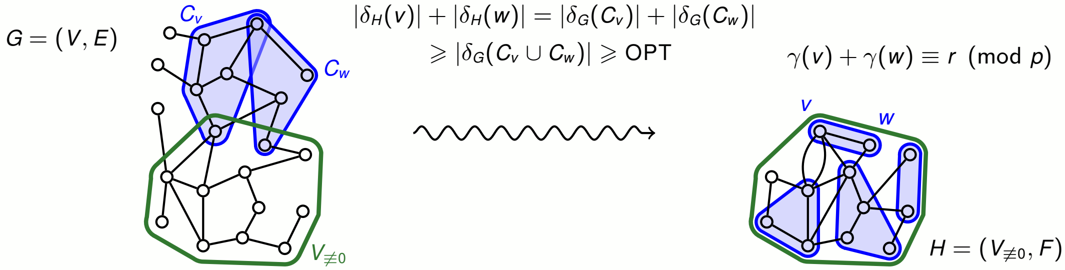
- Combine singletons to $\frac{1}{p}|V_{\not\equiv 0}|$ many feasible sets.
  $\rightsquigarrow$ $\sum_{v \in V_{\not\equiv 0}} |\delta_H(v)| \geqslant \frac{1}{p} \cdot |V_{\not\equiv 0}| \cdot \mathrm{OPT}$.

CCMC with Prime Modulus p

- **Issue:** Singletons in H do not necessarily correspond to cuts with $\gamma(C) \equiv r \pmod{p}$.

$$|\delta_H(v)| + |\delta_H(w)| = |\delta_G(C_v)| + |\delta_G(C_w)|$$
$$\geqslant |\delta_G(C_v \cup C_w)| \geqslant \mathrm{OPT}$$

$$\gamma(v) + \gamma(w) \equiv r \pmod{p}$$



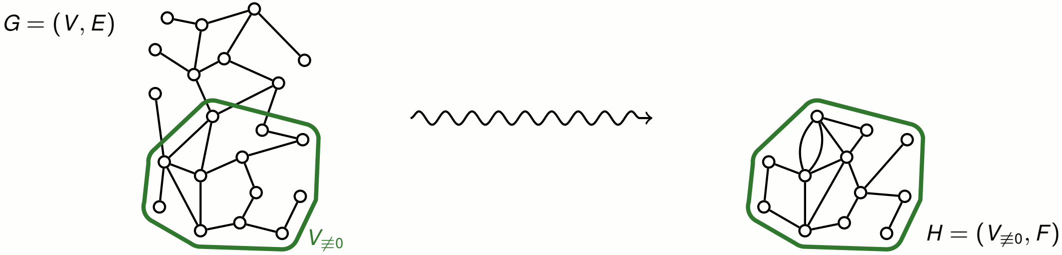
Thm. **(Cauchy-Davenport)** Among any p nonzero elements of $\mathbb{Z}/p\mathbb{Z}$, there is a subset summing to $r \pmod{p}$.

- Combine singletons to $\frac{1}{p}|V_{\neq 0}|$ many feasible sets.
  $\rightsquigarrow \sum_{v \in V_{\neq 0}} |\delta_H(v)| \geqslant \frac{1}{p} \cdot |V_{\neq 0}| \cdot \mathrm{OPT}$.

$\Longrightarrow$ Karger-type average-degree analysis with respect to $V_{\neq 0}$ works!

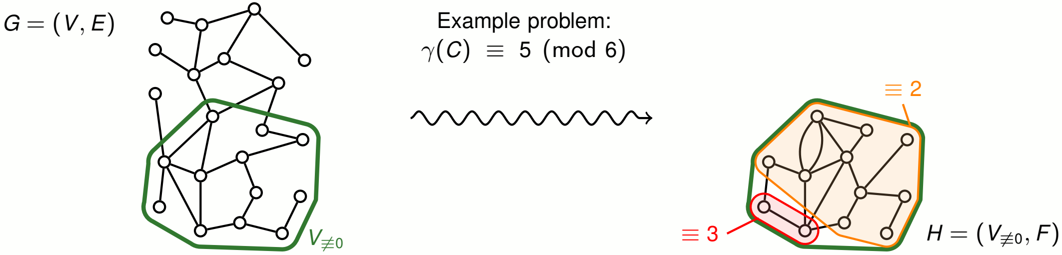
The General Case: Reduction Steps

- **Issue:** We might have $\sum_{v \in V_{\neq 0}} |\delta_H(v)| < \varepsilon \cdot |V_{\neq 0}| \cdot \mathrm{OPT}$.



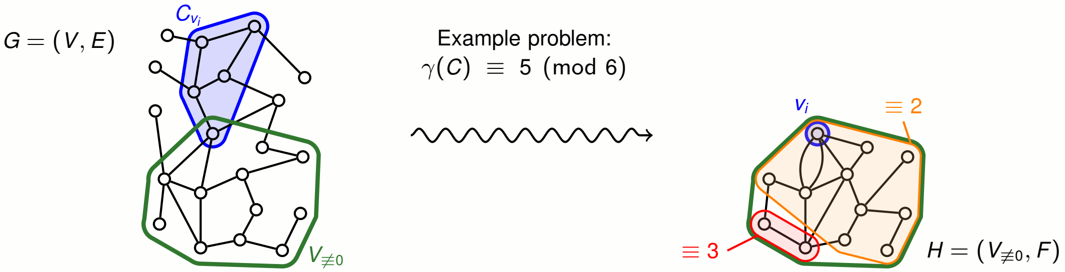
The General Case: Reduction Steps

- **Issue:** We might have $\sum_{v \in V_{\not\equiv 0}} |\delta_H(v)| < \varepsilon \cdot |V_{\not\equiv 0}| \cdot \mathrm{OPT}$.



The General Case: Reduction Steps

- **Issue:** We might have $\sum_{v \in V_{\neq 0}} |\delta_H(v)| < \varepsilon \cdot |V_{\neq 0}| \cdot \mathrm{OPT}$.



- There is $q \in [m-1]$ and many vertices $v_i \in V_{\neq 0}$ with

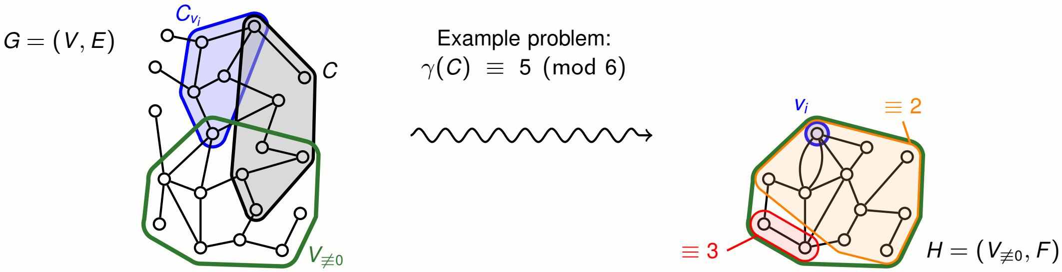
$$|\delta_H(v_i)| < 2\varepsilon\,\mathrm{OPT}$$
$$\text{and} \quad \gamma(v_i) \equiv q \pmod m.$$

- For any cut C, we get

$$|\delta_G(C \,\Delta\, C_{v_i})| < |\delta_G(C)| + 2\varepsilon\,\mathrm{OPT}$$
$$\text{and} \quad \gamma(C \,\Delta\, C_{v_i}) \equiv \gamma(C) \pm q \pmod m.$$

The General Case: Reduction Steps

- **Issue:** We might have $\sum_{v \in V_{\not\equiv 0}} |\delta_H(v)| < \varepsilon \cdot |V_{\not\equiv 0}| \cdot \mathrm{OPT}$.



- There is $q \in [m-1]$ and many vertices $v_i \in V_{\not\equiv 0}$ with

$$|\delta_H(v_i)| < 2\varepsilon\,\mathrm{OPT}$$
$$\text{and} \quad \gamma(v_i) \equiv q \pmod m.$$

- For any cut C, we get

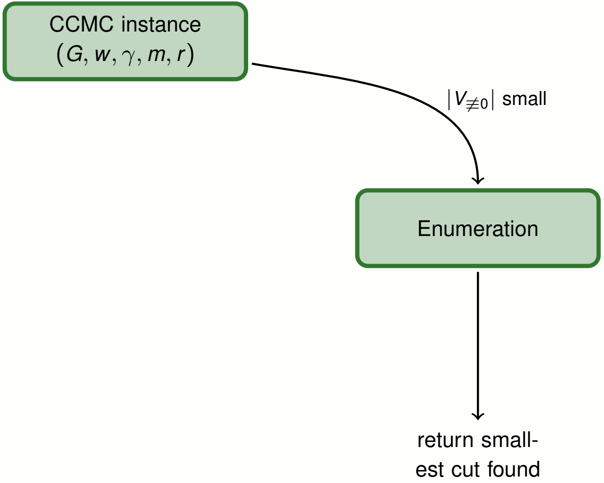
$$|\delta_G(C \,\Delta\, C_{v_i})| < |\delta_G(C)| + 2\varepsilon\,\mathrm{OPT}$$
$$\text{and} \quad \gamma(C \,\Delta\, C_{v_i}) \equiv \gamma(C) \pm q \pmod m.$$

$\Longrightarrow$ Cheap residue correction by multiples of q—leaves problem modulo $\gcd(m, q)$.

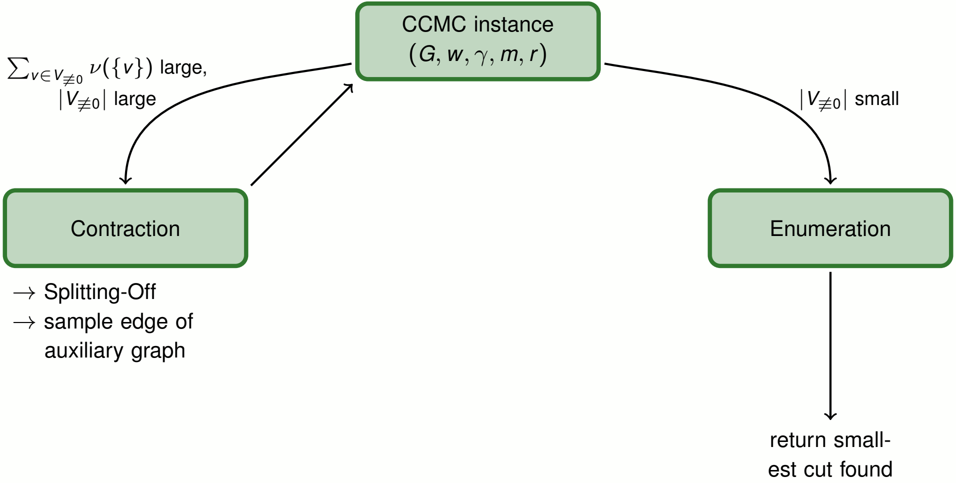
The Complete Algorithm

CCMC instance
(G, w, γ, m, r)

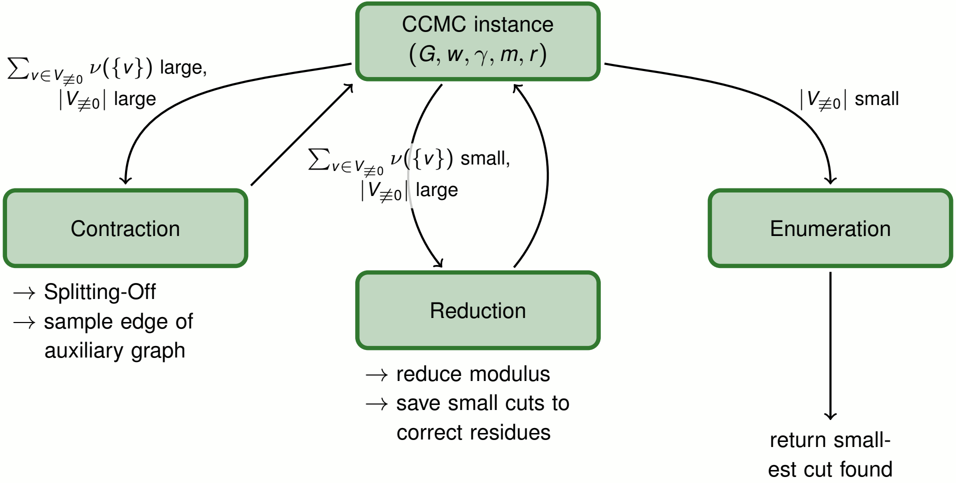
The Complete Algorithm



The Complete Algorithm



The Complete Algorithm



The Complete Algorithm