Network Resilience and the Length-Bounded Multicut Problem: Reaching the Dynamic Billion-Scale with Guarantees

Alan Kuhnle[1,*], Victoria G. Crawford[1], My T. Thai[1]

[1]Computer and Information Science and Engineering, University of Florida, Gainesville, FL
* kuhnle@ufl.edu

ACM SIGMETRICS 2018

- LB MULTICUT: Given a network and a set of pairs of vertices, identify the minimum-size set of vertices (edges) whose removal will sufficiently separate each pair

[1] A. Kuhnle, T. Pan, V. G. Crawford, M. A. Alim, M. T. Thai. Pseudo-Separation for Assessment of Structural Vulnerability of a Network. *Extended Abstract (Poster) in ACM SIGMETRICS*, 2017.

- LB MULTICUT: Given a network and a set of pairs of vertices, identify the minimum-size set of vertices (edges) whose removal will sufficiently separate each pair
- "sufficiently separate" $\rightarrow d(s,t) > T$

[1] A. Kuhnle, T. Pan, V. G. Crawford, M. A. Alim, M. T. Thai. Pseudo-Separation for Assessment of Structural Vulnerability of a Network. *Extended Abstract (Poster) in ACM SIGMETRICS*, 2017.

- LB MULTICUT: Given a network and a set of pairs of vertices, identify the minimum-size set of vertices (edges) whose removal will sufficiently separate each pair
- "sufficiently separate" $\rightarrow d(s, t) > T$
- Motivation: Network robustness

[1] A. Kuhnle, T. Pan, V. G. Crawford, M. A. Alim, M. T. Thai. Pseudo-Separation for Assessment of Structural Vulnerability of a Network. *Extended Abstract (Poster) in ACM SIGMETRICS*, 2017.

- LB MULTICUT: Given a network and a set of pairs of vertices, identify the minimum-size set of vertices (edges) whose removal will sufficiently separate each pair
- "sufficiently separate" $\rightarrow d(s, t) > T$
- Motivation: Network robustness
- Distance as proxy for network functionality[1]

[1] A. Kuhnle, T. Pan, V. G. Crawford, M. A. Alim, M. T. Thai. Pseudo-Separation for Assessment of Structural Vulnerability of a Network. *Extended Abstract (Poster) in ACM SIGMETRICS*, 2017.

- LB MULTICUT: Given a network and a set of pairs of vertices, identify the minimum-size set of vertices (edges) whose removal will sufficiently separate each pair
- "sufficiently separate" $\rightarrow d(s, t) > T$
- Motivation: Network robustness
- Distance as proxy for network functionality[1]
- For example, distance could encode packet loss in a communication network or travel time in a road network

[1] A. Kuhnle, T. Pan, V. G. Crawford, M. A. Alim, M. T. Thai. Pseudo-Separation for Assessment of Structural Vulnerability of a Network. *Extended Abstract (Poster) in ACM SIGMETRICS*, 2017.

- LB MULTICUT: Given a network and a set of pairs of vertices, identify the minimum-size set of vertices (edges) whose removal will sufficiently separate each pair
- "sufficiently separate" $\rightarrow d(s,t) > T$
- Motivation: Network robustness
- Distance as proxy for network functionality[1]
- For example, distance could encode packet loss in a communication network or travel time in a road network
- T can be input (LB MULTICUT) or fixed (T-MULTICUT)

[1] A. Kuhnle, T. Pan, V. G. Crawford, M. A. Alim, M. T. Thai. Pseudo-Separation for Assessment of Structural Vulnerability of a Network. *Extended Abstract (Poster) in ACM SIGMETRICS*, 2017.

Vulnerability assessment: LB MULTICUT

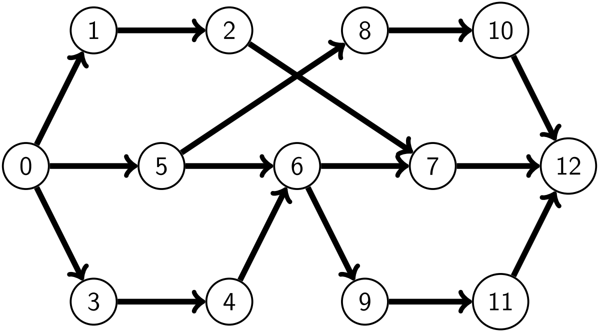

Figure: $\mathcal{S} = \{(0, 12)\}, \; T = 5.$

Vulnerability assessment: LB MULTICUT

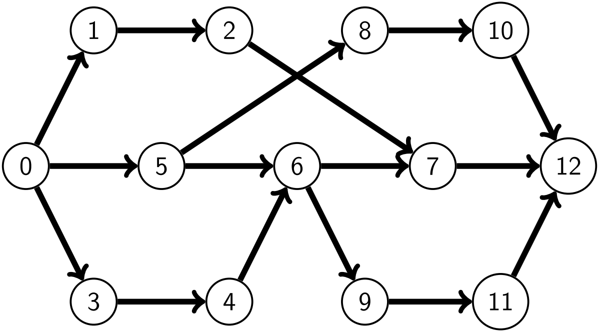

Figure: Want: min-size set of edges to remove s.t. $d(0, 12) > 5$

Vulnerability assessment: LB MULTICUT

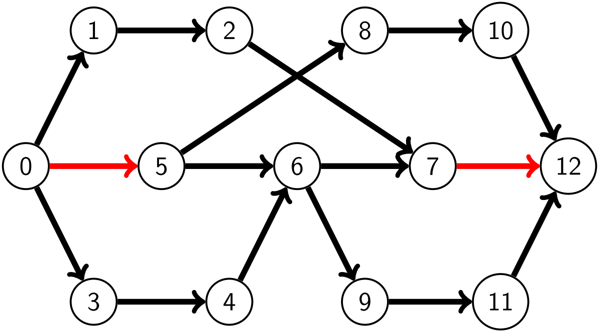

Figure: Optimal solution has 2 edges.

Vulnerability assessment: LB MULTICUT

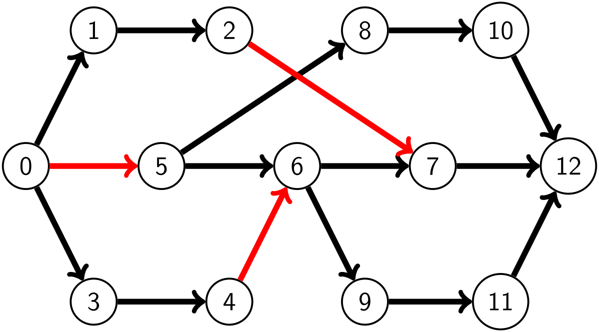

Figure: Classical cut of (s, t) must take three edges

Vulnerability assessment: pseudo-separation

For multicut, vertex and edge versions are not equivalent.

Figure: Approximation results

Problem	Undirected	Directed
CUT (both)	1[2]	1
MULTICUT (edge)	$O(\log k)$[3]	$O(n^{11/23})$[4]
P. MULTICUT (edge)	$O(\log^2 n \log\log n)$[5]	-
MULTICUT (vertex)	$O(\log k)$[6]	-
T-CUT (both)	$T/2$[7]	$T/2$[7]
T-MULTICUT (both)	$T+1$[8]	$T+1$[8]
P. T-MULTICUT (both)	$(T+1)/\epsilon$[8]	$(T+1)/\epsilon$[8]

[2] Elias, Feinstein, and Shannon; Ford, Fulkerson (1956)
[3] Leighton and Rao (1988) [1]
[4] Agarwal, Alon, Charikar (2007) [2]
[5] Konemann, Parekh, and Segev (2006) [3]
[6] Garg, Vazirani, and Yannakakis (1994) [4]
[7] Baier et al. (2010) [5]
[8] Kuhnle et al. (2017) [6]

- Results of Kuhnle et al. (2017):

- Results of Kuhnle et al. (2017):
  - Formulated vulnerability assessment based upon LB MULTICUT

- Results of Kuhnle et al. (2017):
  - Formulated vulnerability assessment based upon LB MULTICUT
  - Provided approximation algorithms when T is fixed parameter

- Results of Kuhnle et al. (2017):
  - Formulated vulnerability assessment based upon LB MULTICUT
  - Provided approximation algorithms when T is fixed parameter
    - Require enumeration of all paths of length at most T

- Results of Kuhnle et al. (2017):
  - Formulated vulnerability assessment based upon LB MULTICUT
  - Provided approximation algorithms when T is fixed parameter
    - Require enumeration of all paths of length at most T
  - Approximability lower bound of $2 - \epsilon$ (assuming UGC)

Contributions

- Primal-dual algorithm

Contributions

- Primal-dual algorithm
    - Approximation ratio: T, even when T is input

Contributions

- Primal-dual algorithm
  - Approximation ratio: T, even when T is input
  - Time complexity:

$$O(\underbrace{k}_{\text{Number of pairs.}} \underbrace{D}_{\text{Max. degree}} \underbrace{(m + n\log(n))}_{\text{Dijkstra's alg.}})$$

Contributions

- Primal-dual algorithm
  - Approximation ratio: T, even when T is input
  - Time complexity:

$$O(\underbrace{k}_{\text{Number of pairs.}} \underbrace{D}_{\text{Max. degree}} \underbrace{(m + n\log(n))}_{\text{Dijkstra's alg.}})$$

- Inapproximability result:

$$\Omega(T),$$

unless $NP \subseteq BPP$.

Contributions

- Primal-dual algorithm
  - Approximation ratio: T, even when T is input
  - Time complexity:

$$O(\underbrace{k}_{\text{Number of pairs.}} \underbrace{D}_{\text{Max. degree}} \underbrace{(m + n\log(n))}_{\text{Dijkstra's alg.}})$$

- Inapproximability result:

$$\Omega(T),$$

  unless $NP \subseteq BPP$.
- Additional approximation algorithms.

Primal-dual approach

- **Primal-dual**

Primal-dual approach

- **Primal-dual**
  - Until feasible,

Primal-dual approach

- **Primal-dual**
  - Until feasible,
  - remove (*i.e.* add to solution) an overall shortest path between any pair in $\mathcal{S}$

Primal-dual approach

- **Primal-dual**
  - Until feasible,
  - remove (*i.e.* add to solution) an overall shortest path between any pair in $\mathcal{S}$
- To improve solution quality in practice, can prune it

Primal-dual approach

- **Primal-dual**
  - Until feasible,
  - remove (*i.e.* add to solution) an overall shortest path between any pair in $\mathcal{S}$
- To improve solution quality in practice, can prune it
- "fully dynamic": solution can be updated upon changes in edge weight and vertex insertion / deletion

Primal-dual, example

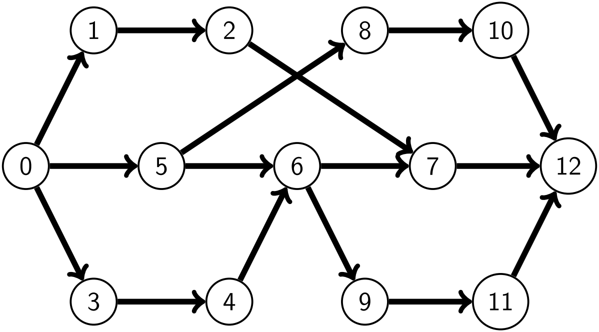

Figure: $d(0, 12) = 4,\ T = 5$

Primal-dual, example

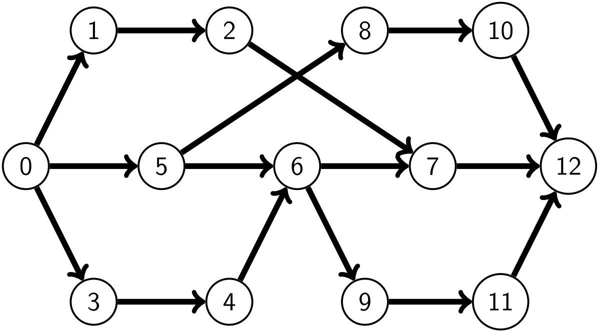

Figure: Compute shortest path and add all edges into solution.

Primal-dual, example

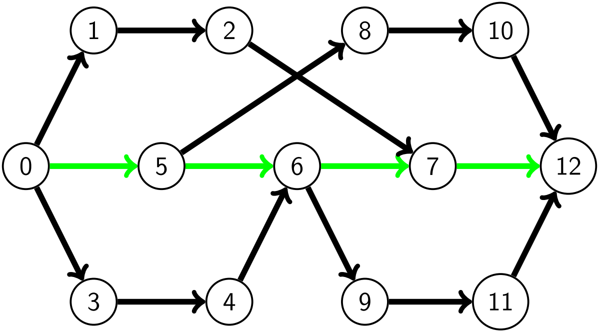

Figure: Compute shortest path and add all edges into solution.

Primal-dual, example

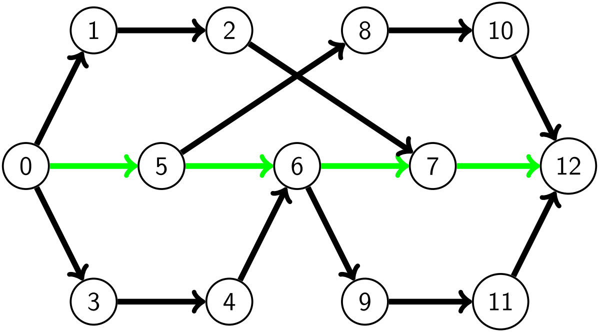

Figure: $d(0, 12) = 6,\ T = 5$

Primal-dual, example

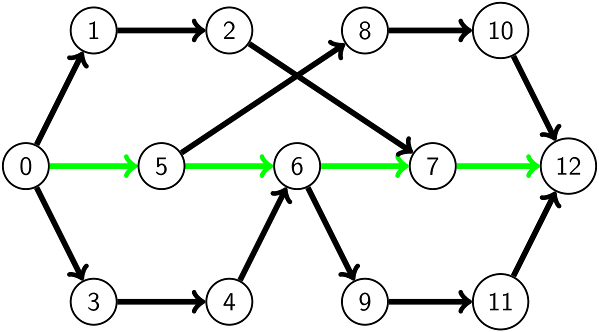

Figure: So we're done.

Primal-dual, example

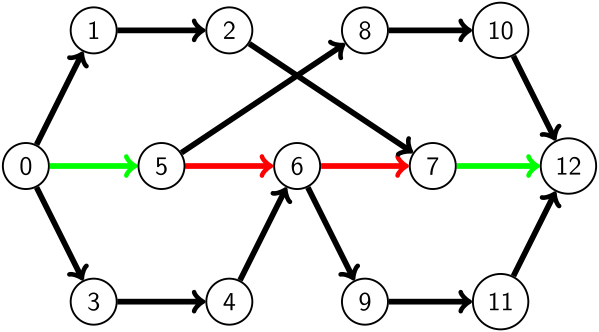

Figure: Can prune edges $(5, 6), (6, 7)$

Primal-dual, example

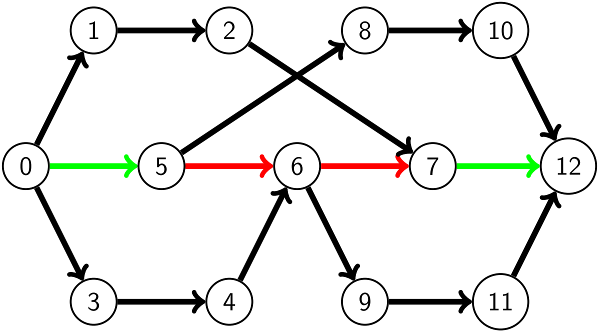

Figure: How to update?

Primal-dual, example

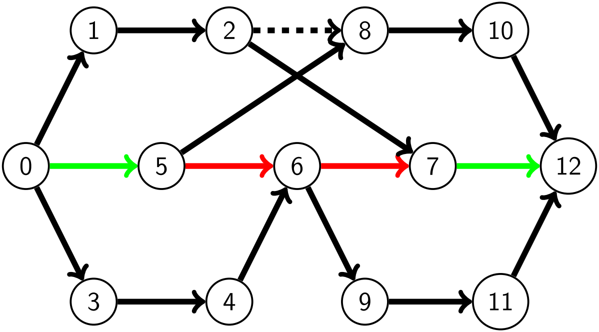

Figure: Add $(2, 8)$

Primal-dual, example

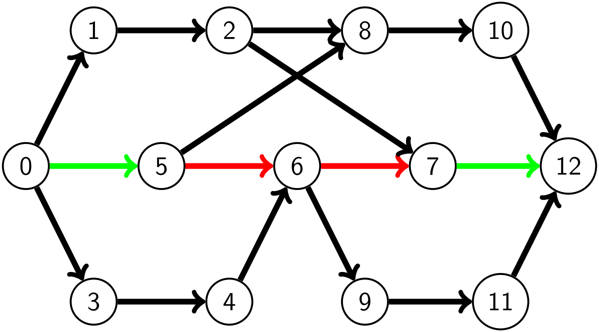

Figure: Add $(2, 8)$

Primal-dual, example

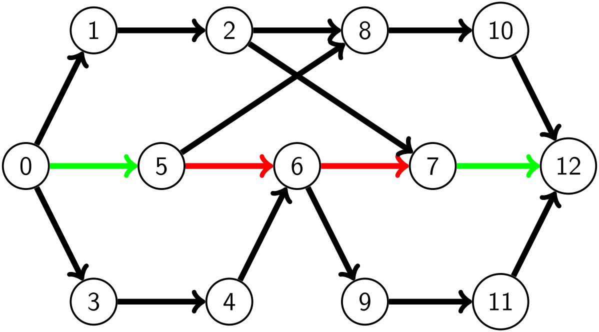

Figure: $d(0, 12) = 5$, $T = 5$

Primal-dual, example

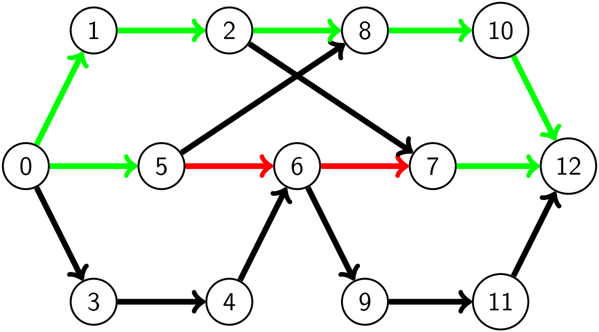

Figure: Add shortest path

Primal-dual, example

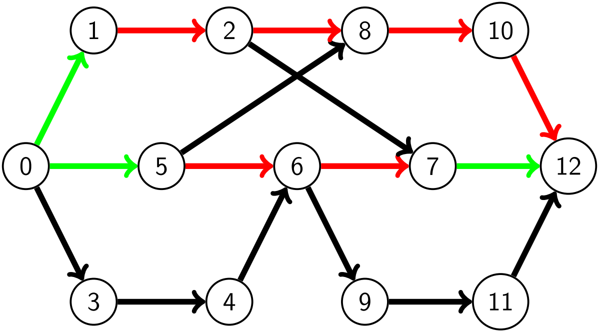

Figure: Prune

TAG Algorithm

Strategy:

- Maintains a maximal, edge-disjoint collection U of paths

TAG Algorithm

Strategy:

- Maintains a maximal, edge-disjoint collection U of paths
- Number of paths lower bounds OPT

TAG Algorithm

Strategy:

- Maintains a maximal, edge-disjoint collection U of paths
- Number of paths lower bounds OPT
- Ratio T

Theorem

Let $T \geq 16$. Unless $NP \subseteq BPP$, there is no polynomial-time algorithm to approximate T-LB MULTICUT within a factor of

$$\left\lfloor \frac{T}{6} \right\rfloor - 1 - \epsilon,$$

for any $\epsilon > 0$.

- Approximation-preserving reduction from Cycle Interdiction Problem: minimum set of edges to intersect all cycles of length at most r. Guruswami et al. showed lower bound of

$$\left\lfloor \frac{r}{2} \right\rfloor - 1 - \epsilon.$$

Theorem

Let $T \geq 16$. Unless $NP \subseteq BPP$, there is no polynomial-time algorithm to approximate T-LB MULTICUT within a factor of

$$\left\lfloor \frac{T}{6} \right\rfloor - 1 - \epsilon,$$

for any $\epsilon > 0$.

- Approximation-preserving reduction from Cycle Interdiction Problem: minimum set of edges to intersect all cycles of length at most r. Guruswami et al. showed lower bound of

$$\left\lfloor \frac{r}{2} \right\rfloor - 1 - \epsilon.$$

- Requires $\Omega(n)$ pairs.

Theorem

Let $T \geq 16$. Unless $NP \subseteq BPP$, there is no polynomial-time algorithm to approximate T-LB MULTICUT within a factor of

$$\left\lfloor \frac{T}{6} \right\rfloor - 1 - \epsilon,$$

for any $\epsilon > 0$.

- Approximation-preserving reduction from Cycle Interdiction Problem: minimum set of edges to intersect all cycles of length at most r. Guruswami et al. showed lower bound of

$$\left\lfloor \frac{r}{2} \right\rfloor - 1 - \epsilon.$$

- Requires $\Omega(n)$ pairs.
- For single pair T-CUT, best lower bound is currently $\sqrt{T}$.

Experimental Evaluation

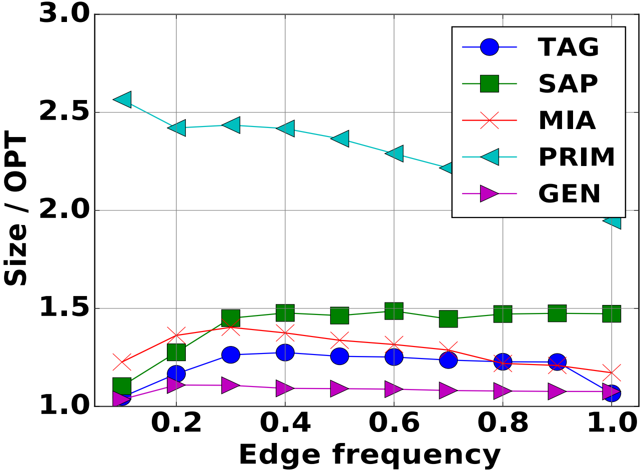

Figure: Static experiments on ER 100 networks.

Experimental Evaluation

Table: Real-world traces.

Network	$\|V\|$	$\|E\|$	$\hat{d}(x, y)$	Weighted
Gnutella	6.301×10^3	2.078×10^4	4.63	No
Enron	3.669×10^4	1.838×10^5	4.04	No
RoadSF	1.748×10^5	2.218×10^5	3635.29	Yes
Google	8.757×10^5	4.322×10^6	6.33	No
Skitter	1.696×10^6	1.109×10^7	13.00	Yes
Friendster	1.248×10^8	1.806×10^9	4.99	No

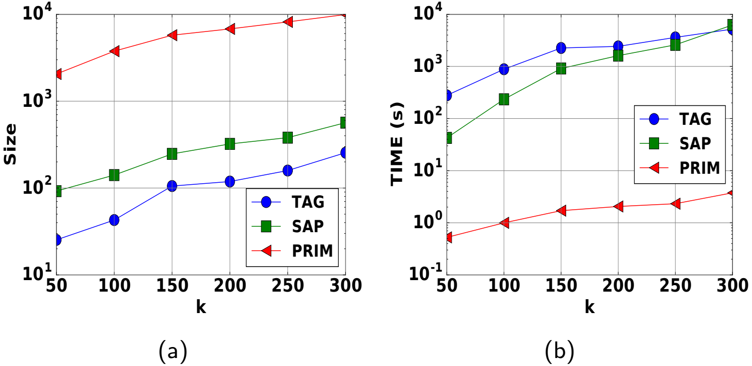

Figure: RoadSF: (a) Solution Size, (b) Running time vs. number of pairs k

Table: Static average add/remove results (TAG).

Dataset	Addition time (s)	Removal time(s)	Addition size	Removal size
Gnutella	0.6	0.2	125.4	30.0
Enron	21.6	3.9	269.8	76.2
Google	3765.9	696.4	617.0	155.4
Friendster	38127.0	32465.0	94.0	49.6

Table: Dynamic average add/remove results (TAG).

Dataset	Addition time (ms)	Removal time (ms)	Addition loss	Removal loss
Gnutella	1.0	1.0	6.7%	0.0%
Enron	3.5	0.7	7.0%	15.5%
Google	110.0	12.0	20.3%	13.3%
Friendster	1100.0	0.2	0.0%	2.8%

Summary

- Scalable and dynamic algorithm with ratio T

Summary

- Scalable and dynamic algorithm with ratio T
- Lower bound of $T/6 - 1$, $T \geq 16$

Summary

- Scalable and dynamic algorithm with ratio T
- Lower bound of $T/6 - 1$, $T \geq 16$
- Implementations available: `https://gitlab.com/kuhnle/multi-pcut`

Summary

- Scalable and dynamic algorithm with ratio T
- Lower bound of $T/6 - 1$, $T \geq 16$
- Implementations available: `https://gitlab.com/kuhnle/multi-pcut`
- Future work: improve lower bound or better approximation?

Thank you! Questions?

Alan Kuhnle, kuhnle@ufl.edu

References I

[1] T. Leighton and S. Rao.
An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms.
[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science, pages 422–431, 1988.

[2] Amit Agarwal, Noga Alon, and Moses S Charikar.
Improved approximation for directed cut problems.
In *Proceedings of the thirty-ninth annual ACM symposium on Theory of Computing*, pages 671–680, New York, NY USA, 2007. ACM.

[3] Jochen Könemann, Ojas Parekh, and Danny Segev.
A unified approach to approximating partial covering problems.
In *European Symposium on Algorithms*, Berlin, 2006. Springer.

[4] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis.
Multiway Cuts in Directed and Node Weighted Graphs.
In *International Colloquium on Automata, Languages, and Programming.*, Berlin, 1994. Springer-Verlag.

[5] Georg Baier, Thomas Erlebach, Alexander Hall, Ekkehard Koehler, Petr Kolman, Ondrej Pangrac, Heiko Schilling, and Martin Skutella.
Length-Bounded Cuts and Flows.
ACM Transactions on Algorithms, 7(1):1–27, 2010.

[6] A. Kuhnle, T. Pan, V.G. Crawford, M.A. Alim, and M.T. Thai.
Pseudo-separation for assessment of structural vulnerability of a network.
In *SIGMETRICS 2017 Abstracts - Proceedings of the 2017 ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems*, 2017.