Impact of Network Coding on Combinatorial Optimization

Chandra Chekuri
Univ. of Illinois, Urbana-Champaign

DIMACS Workshop on Network Coding: Next 15 Years
December 16, 2015

Network Coding

[Ahlswede-Cai-Li-Yeung]

Beautiful result that established connections between

- Coding and communication theory
- Networks and graphs
- Combinatorial Optimization
- Many others …

Combinatorial Optimization

“Good characterizations” via “Min-Max” results is key to algorithmic success

Multicast network coding result is a min-max result

Benefits *to* Combinatorial Optimization

My perspective/experience

- New applications of existing results
- New problems
- New algorithms for classical problems
- Challenging open problems
- Interdisciplinary collaborations/friendships

Outline

- **Part 1:** Quantifying the benefit of network coding over routing
- **Part 2:** Algebraic algorithms for connectivity

Part 1

Coding Advantage

Question: What is the *advantage* of network coding in improving throughput over routing?

Coding Advantage

Question: What is the *advantage* of network coding in improving throughput over routing?

Motivation

- Basic question since routing is standard and easy
- To understand and approximate capacity

Different Scenarios

- Unicast in wireline zero-delay networks
- Multicast in wireline zero-delay networks
- Multiple unicast in wireline zero-delay networks
- Broadcast/wireless networks
- Delay constrained networks

Undirected graphs vs *directed* graphs

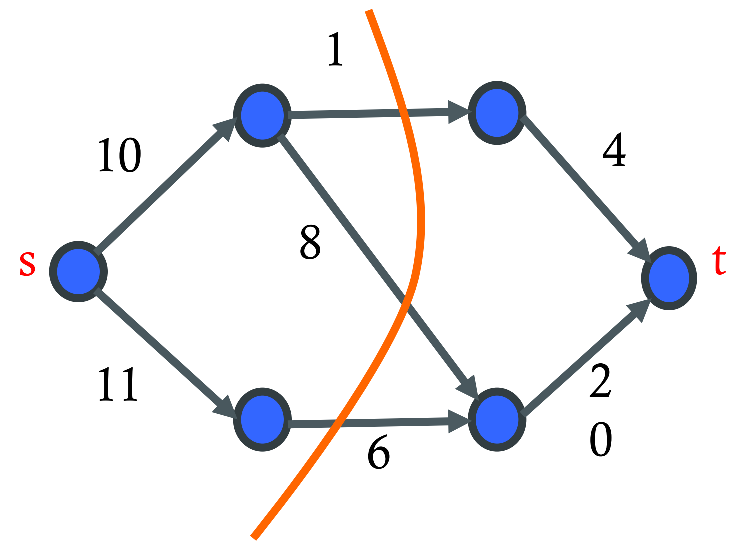
Max-flow Min-cut Theorem

[Ford-Fulkerson, Menger]

G=(V,E) directed graph with non-negative edge-capacities

max s-t flow value equal to min s-t cut value

if capacities *integral* max flow can be chosen to be *integral*



Min s-t cut value upper bound on information capacity

No coding advantage

Edmonds Arborescence Packing Theorem

[Edmonds]

G=(V,E) directed graph with non-negative edge-capacities

A s-arboresence is a out-tree T rooted at s that contains all nodes in V

Theorem: There are k edge-disjoint s-arborescences in G if and only if the s-v mincut is k for all v in V

Min s-t cut value upper bound on information capacity

No coding advantage for *multicast* from s to all nodes in V

Enter Network Coding

Multicast from s to a *subset* of nodes T

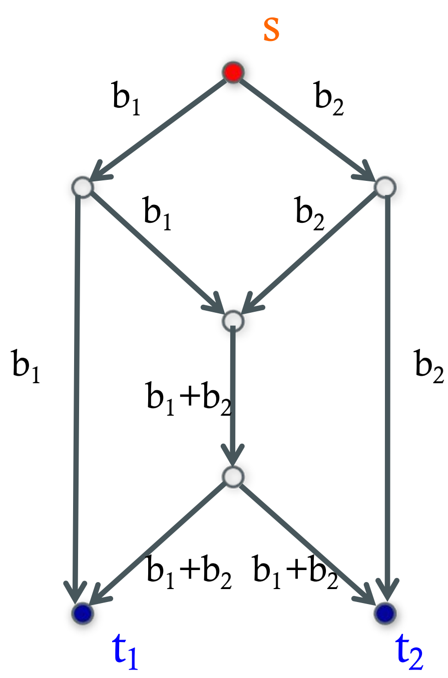
[Ahlswede-Cai-Li-Yeung]

Theorem: Information capacity is equal to min cut from s to a terminal in T

What about routing? Packing Steiner trees

How big is the coding advantage?

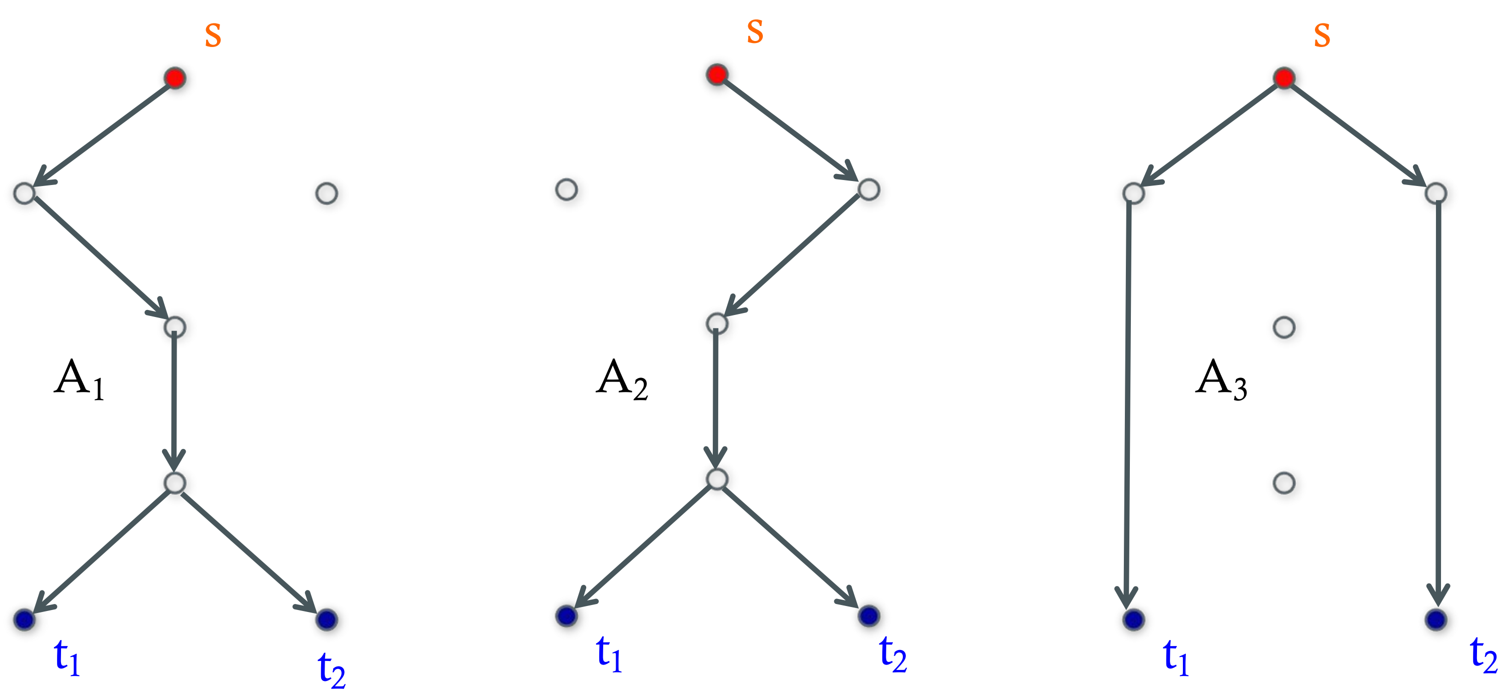
Multicast Example



s can multicast to t_1 and t_2 at *rate* 2 using network coding

Optimal rate since
min-cut(s, t_1) = *min-cut*(s, t_2) = 2

Question: what is the best achievable rate without coding (only routing) ?



A_1, A_2, A_3 are multicast/Steiner trees: each edge of G in at most 2 trees
Use each tree for ½ the time. Rate = 3/2

Packing Steiner trees

Question: If mincut from s to each t in T is k, how many Steiner trees can be packed?

- Packing questions fundamental in combinatorial optimization
- Optimum packing can be written as a "big" LP
- Connected to several questions on Steiner trees

Several results/connections

- [Li, Li] In undirected graphs coding advantage for multicast is at most 2
- [Agarwal-Charikar] In undirected graphs coding advantage for multicast is *exactly* equal to the integrality gap of the bi-directed relaxation for Steiner tree problem. Gap is at most 2 and at least 8/7. An important **unresolved** problem in approximation.
- [Agarwal-Charikar] In directed graphs coding advantage is *exactly* equal to the integrality gap of the natural LP for directed Steiner tree problem. Important **unresolved** problem. Via results from [Zosin-Khuller, Halperin etal] coding advantages is $\Omega(k^{1/2})$ or $\Omega(\log^2 n)$
- [C-Fragouli-Soljanin] extend results to lower bound coding advantage for *average* throughput and heterogeneous settings

New Theorems

[Kiraly-Lau’06]

“Approximate min-max theorems for Steiner rooted-orientation of graphs and hypergraphs”

[FOCS’06, Journal of Combinatorial Theory ‘08]

Motivated directly by network coding for multicast

Multiple Unicast

$G=(V,E)$ and multiple pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$

What is the coding advantage for multiple unicast?

- In directed graphs it can be $\Omega(k)$ [Harvey etal]
- In undirected graphs it is unknown! [Li-Li] conjecture states that there is no coding advantage

Multiple Unicast

What is the coding advantage for multiple unicast?

- Can be *upper bounded* by the gap between **maximum concurrent flow** and **sparsest cut**
- Extensive work in theoretical computer science
- Many results known

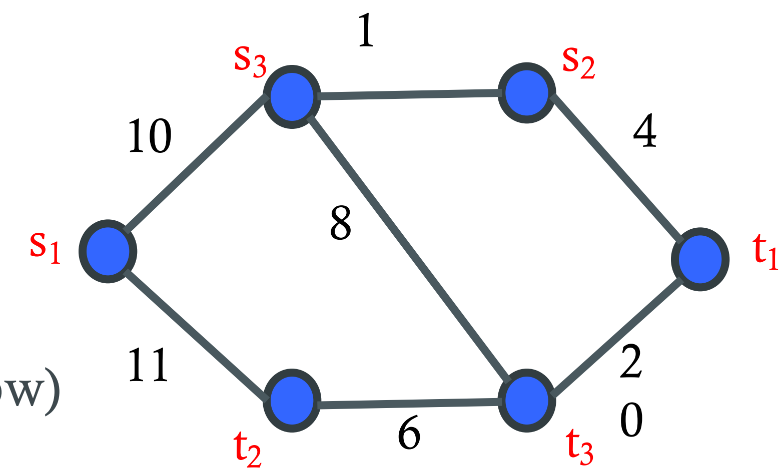
Max Concurrent Flow and Min Sparsest Cut

$f_i(e)$: flow for pair i on edge e

$\sum_i f_i(e) \leq c(e)$ for all e

$val(f_i) \geq \lambda D_i$ for all i

$\max \lambda$ (max concurrent flow)



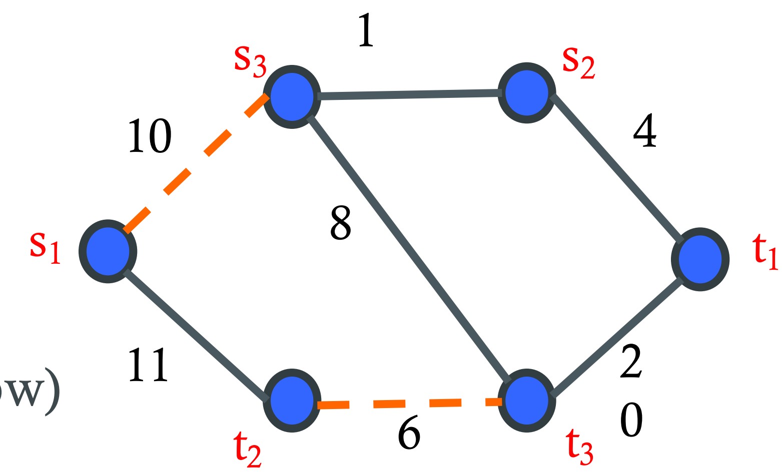
Max Concurrent Flow and Min Sparsest Cut

$f_i(e)$: flow for pair i on edge e

$\sum_i f_i(e) \leq c(e)$ for all e

$val(f_i) \geq \lambda D_i$ for all i

max λ (max concurrent flow)



Sparsity of cut = capacity of cut / demand separated by cut

Max Concurrent Flow $\leq$ Min Sparsity

Known Flow-Cut Gap Results

Scenario	Flow-Cut Gap
Undirected graphs	$\Theta(\log k)$
Directed graphs	$O(k)$, $O(n^{11/23})$, $\Omega(k)$, $\Omega(n^{1/7})$
Directed graphs, *symmetric demands*	$O(\log k \log \log k)$, $\Omega(\log k)$

Symmetric Demands

$G=(V,E)$ and multiple pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$

s_i wants to communicate with t_i **and** t_i wants to communicate with s_i *at the same rate*

[Kamath-Kannan-Viswanath] showed that flow-cut gap translates to upper bound on coding advantage. Using GNS cuts

Challenging Questions

How to understand capacity?

- [Li-Li] conjecture and understanding gap between flow and capacity in undirected graphs
- Can we obtain a *slightly non-trivial* approximation to capacity in directed graphs?

Capacity of Wireless Networks

Capacity of wireless networks

Major issues to deal with:

- interference due to broadcast nature of medium
- noise

Capacity of wireless networks

Understand/model/approximate wireless networks via wireline networks

- Linear deterministic networks [Avestimehr-Diggavi-Tse'09]
  - *Unicast/multicast (single source).* Connection to polylinking systems and submodular flows [Amaudruz-Fragouli'09, Sadegh Tabatabaei Yazdi-Savari'11, Goemans-Iwata-Zenklusen'09]
- Polymatroidal networks [Kannan-Viswanath'11]
  - *Multiple unicast.*

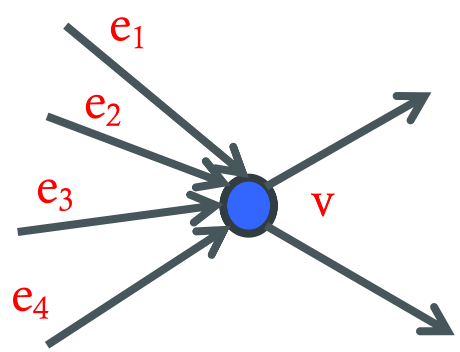
Key to Success

Flow-cut gap results for polymatroidal networks

- Originally studied by [Edmonds-Giles] (submodular flows) and [Lawler-Martel] for single-commodity
- More recently for multicommodity [C-Kannan-Raja-Viswanath'12] motivated by questions from models of [Avestimehr-Diggavi-Tse'09] and several others

Polymatroidal Networks

Capacity of edges incident to v *jointly constrained* by a polymatroid (monotone non-neg submodular set func)



$\sum_{i \in S} c(e_i) \leq f(S)$ for every $S \subseteq \{1,2,3,4\}$

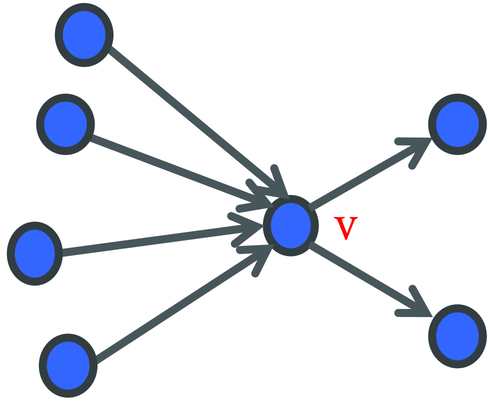
Directed Polymatroidal Networks

[Lawler-Martel'82, Hassin'79]

Directed graph $G=(V,E)$

For each node v two polymatroids

- ρ_v^- with ground set $\delta^-(v)$
- ρ_v^+ with ground set $\delta^+(v)$

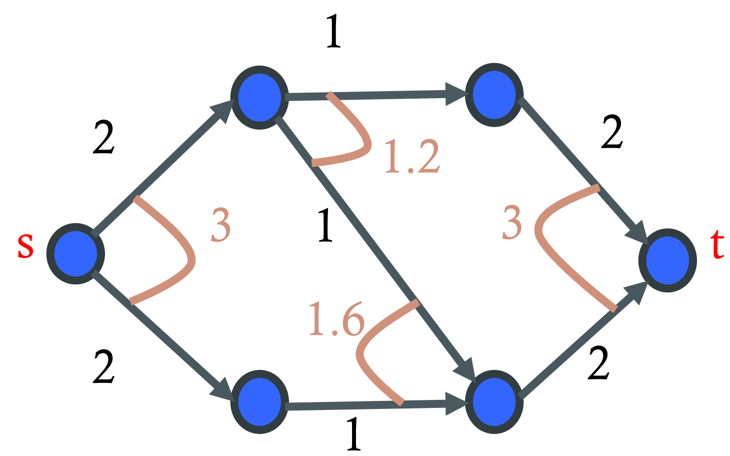


$$\sum_{e \in S} f(e) \leq \rho_v^-(S) \text{ for all } S \subseteq \delta^-(v)$$

$$\sum_{e \in S} f(e) \leq \rho_v^+(S) \text{ for all } S \subseteq \delta^+(v)$$

s-t flow

Flow from s to t: “standard flow” with polymatroidal capacity constraints



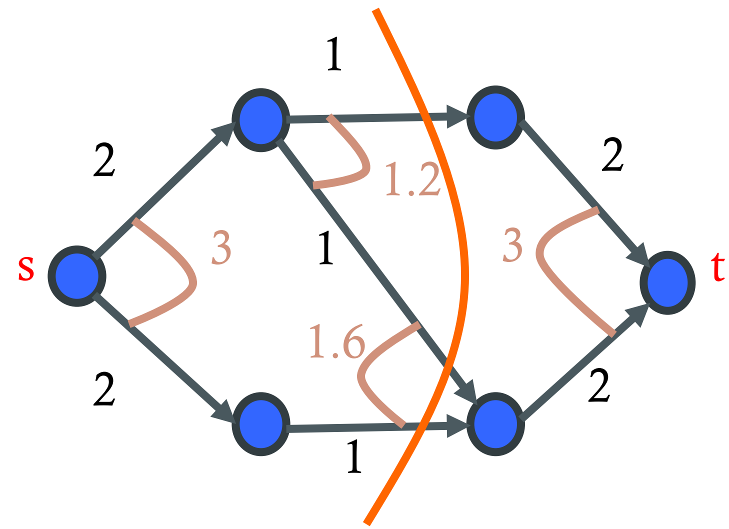
What is the cap. of a cut?

Assign each edge (a,b) of cut to either a or b

Value = sum of function values on assigned sets

Optimize over all assignments

min{1+1+1, 1.2+1, 1.6+1}



Maxflow-Mincut Theorem

[Lawler-Martel'82, Hassin'79]

Theorem: In a directed polymatroidal network the max s-t flow is equal to the min s-t cut value.

Model equivalent to submodular-flow model of[Edmonds-Giles'77] that can derive as special cases

- polymatroid intersection theorem
- maxflow-mincut in standard network flows
- Lucchesi-Younger theorem

Multi-commodity Flows

Polymatroidal network $G=(V,E)$

k pairs $(s_1,t_1),\ldots,(s_k,t_k)$

Multi-commodity flow:

- f_i is s_i-t_i flow
- $f(e) = \sum_i f_i(e)$ is total flow on e
- flows on edges constrained by polymatroid constraints at nodes

Multi-commodity Cuts

Polymatroidal network $G=(V,E)$

k pairs $(s_1,t_1),\ldots,(s_k,t_k)$

Multicut: set of edges that separates all pairs

Sparsity of cut: cost of cut/demand separated by cut

Cost of cut: as defined earlier via optimization

Main Result

[C-Kannan-Raja-Viswanath'12]

Flow-cut gaps for polymatroidal networks essentially match the known bounds for standard networks

Scenario	Flow-Cut Gap
Undirected graphs	$\Theta(\log k)$
Directed graphs	$O(k)$, $O(n^{11/23})$, $\Omega(k)$, $\Omega(n^{1/7})$
Directed graphs, *symmetric demands*	$O(\log k \log \log k)$, $\Omega(\log k)$

Implications for network information theory

Results on polymatroidal networks and special cases have provided **approximate** understanding of the capacity of a class of wireless networks

Implications for Combinatorial Optimization

- Motived study of multicommodity polymatroidal networks
- Resulted in new results and new proofs of old results
- Several important technical connections bridging submodular optimization and embeddings techniques for flow-cut gap results

Additional work [Lee-Mohorrami-Mendel'14] motivated by questions from polymatroidal networks

Networks with Delay

[Wang-Chen'14] Coding provides constant factor advantage over routing even for unicast!

How much?

Networks with Delay

[Wang-Chen'14] Coding provides constant factor advantage over routing even for unicast!

How much?

[C-Kamath-Kannan-Viswanath'15] At most O(log D)

See Sudeep's talk later in workshop

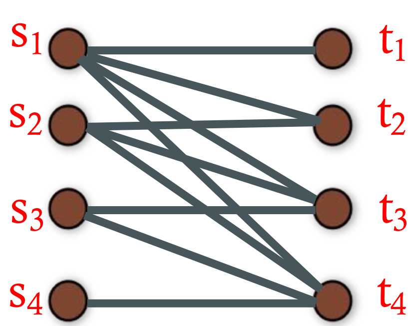
Connections to Combinatorial Optimization

Work in [C-Kamath-Kannan-Viswanath'15] raised a very nice new flow-cut gap problem

"Triangle Cast"

Triangle Cast

Given $G=(V,E)$ terminals $s_1, s_2, \ldots, s_k$ and $t_1, t_2, \ldots, t_k$ communication pattern is s_i to t_j for all $j \geq i$



Connections to Combinatorial Optimization

Work in [C-Kamath-Kannan-Viswanath'15] raised a very nice new flow-cut gap problem

"Triangle Cast"

- Connected to several classical problems such multiway cut, multicut and feedback problems
- Seems to require new techniques to solve
- Inspired several new results [C-Madan'15]

Part 2

Algebraic algorithms for connectivity

Graph Connectivity

- Given a *simple directed* graph G=(V,E) and two nodes s and t, compute the maximum number of edge disjoint paths between s and t.
- Equivalently the min s-t cut value

Fundamental algorithmic problem in combinatorial optimization

Known Algorithms

- [Even-Tarjan'75] $O(\min\{m^{1.5}, n^{2/3}m\})$ run-time, where n is the number of vertices and m is the number of edges.

Recent breakthroughs (ignoring log factor)

- [Madry'13] $O(m^{10/7})$
- [Sidford-Lee'14] $O(mn^{1/2})$

All Pairs Edge Connectivities

- Given simple directed graph G=(V,E) compute s-t edge connectivity for *each pair* (s,t) in V x V
- Not known how to do faster than computing each pair separately. Even from a single source s to all v
- *Undirected* graphs have much more structure. Can compute all pairs in O(mn polylog(n)) time

New Algebraic Approach

[Cheung-Kwok-Lau-Leung'11]

Faster algorithms for connectivity via

"random network coding"

Next few slides from Lap Chi Lau: used with his permission

Random Linear Network Coding

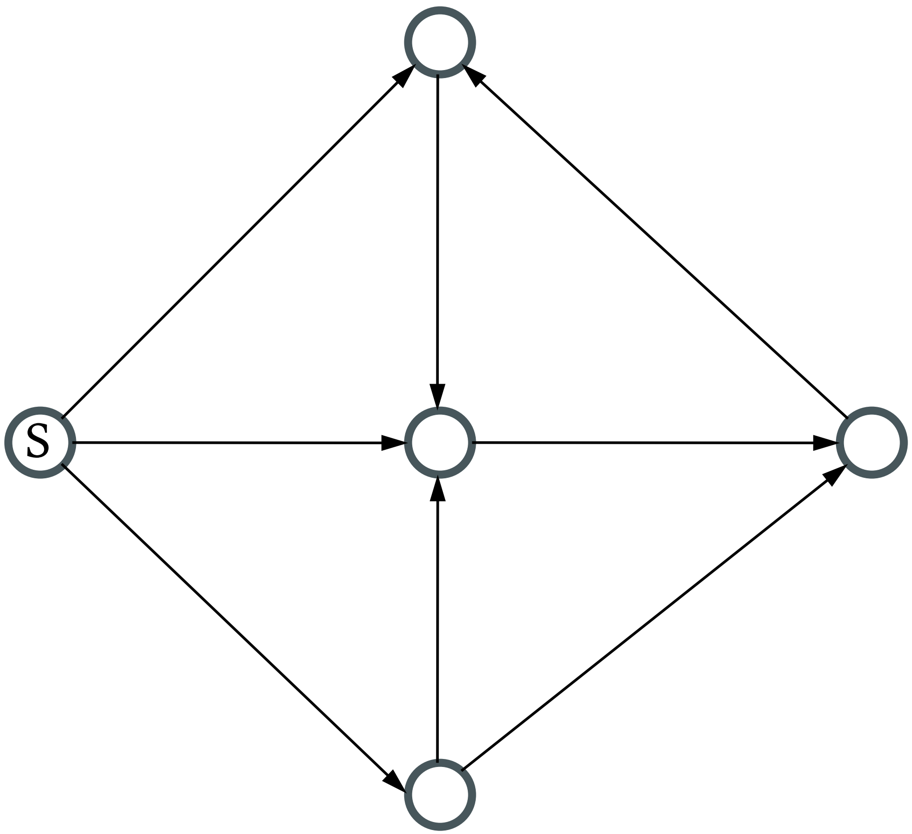
- Random linear network coding is oblivious to network

- [Jaggi] observed that edge connectivity from the source can be determined by looking at the rank of the receiver's vectors. Restricted to *directed acyclic graphs.*

- For general graphs, network coding is more complicated as it requires convolution codes.

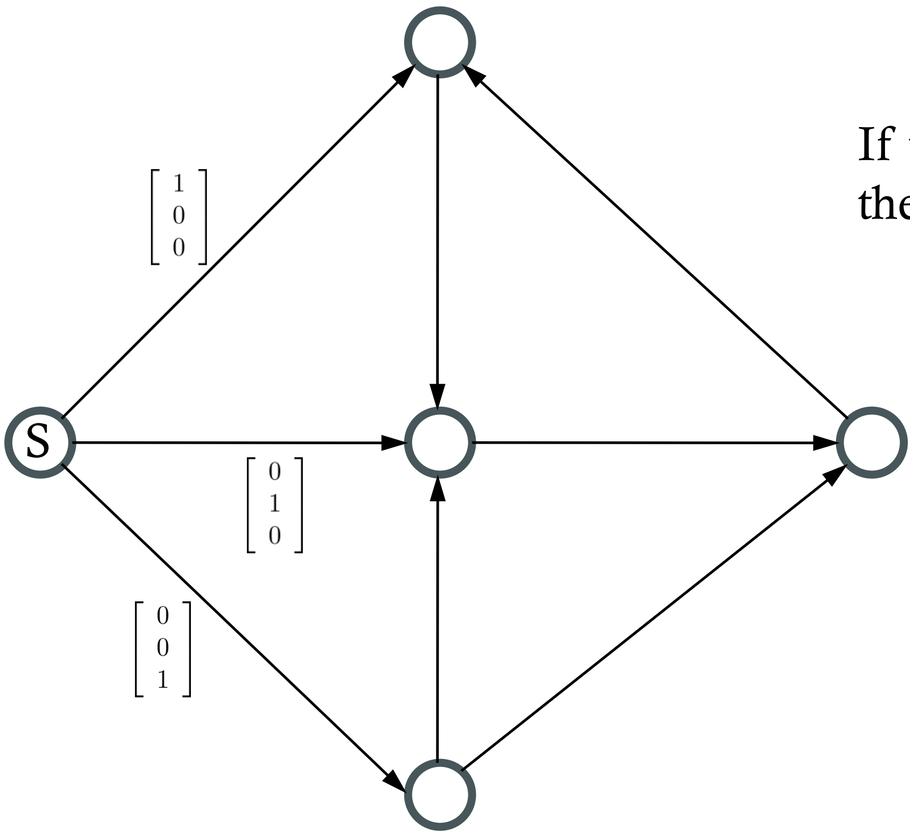
New Algebraic Formulation

Very similar to random linear network coding



New Algebraic Formulation

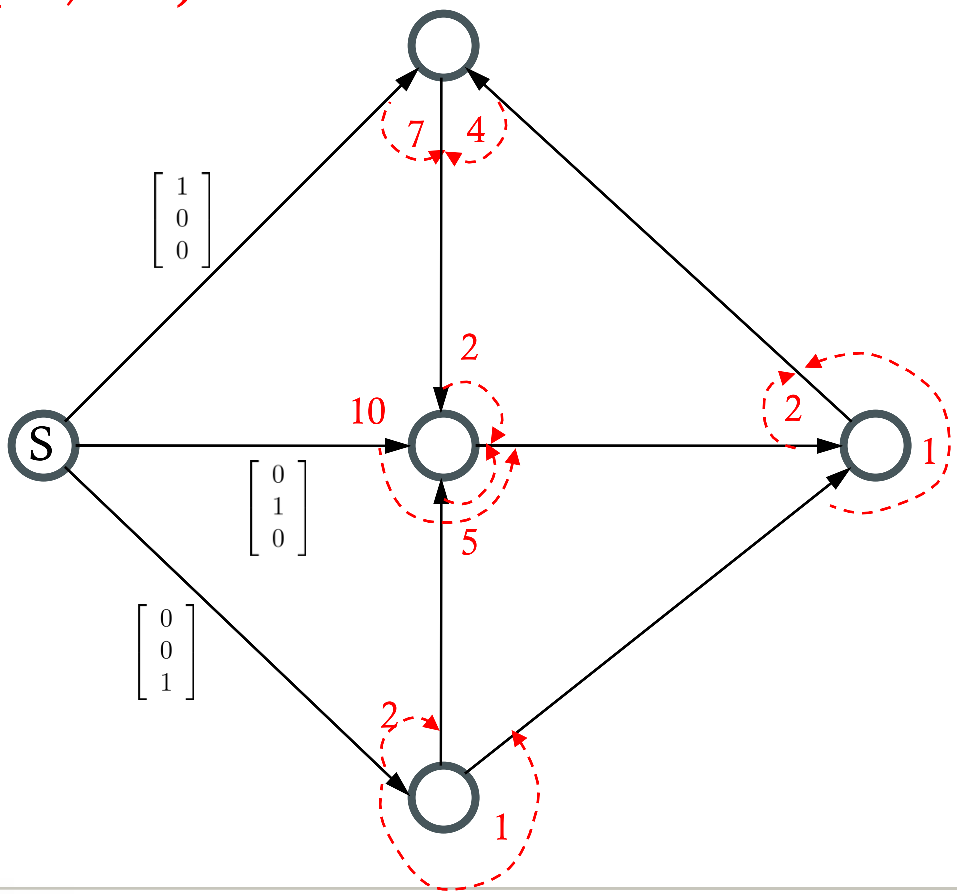
(1) Source sends out linearly independent vectors.



If the source has outdegree d, then the vectors are d-dimensional.

New Algebraic Formulation

(2) Pick *random coefficients* for each pair of adjacent edges (uv, vw)

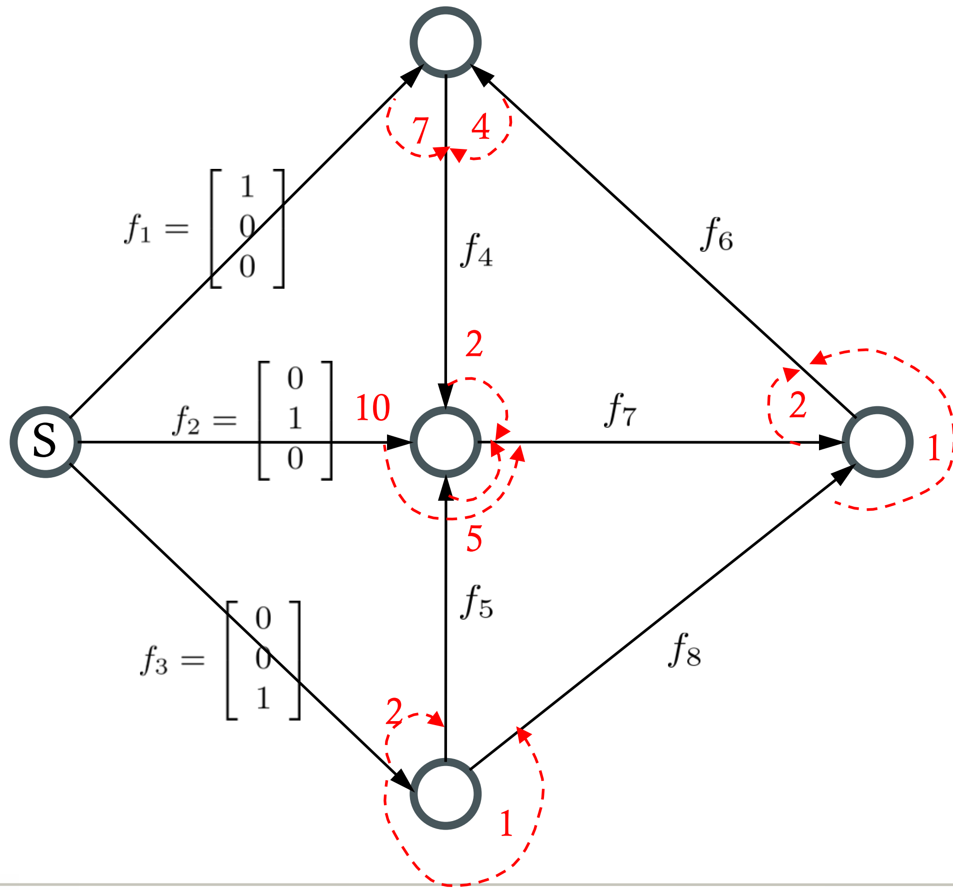


Random coefficients

Field size = 11

New Algebraic Formulation

(3) Require each vector to be a linear combination of its incoming vectors.



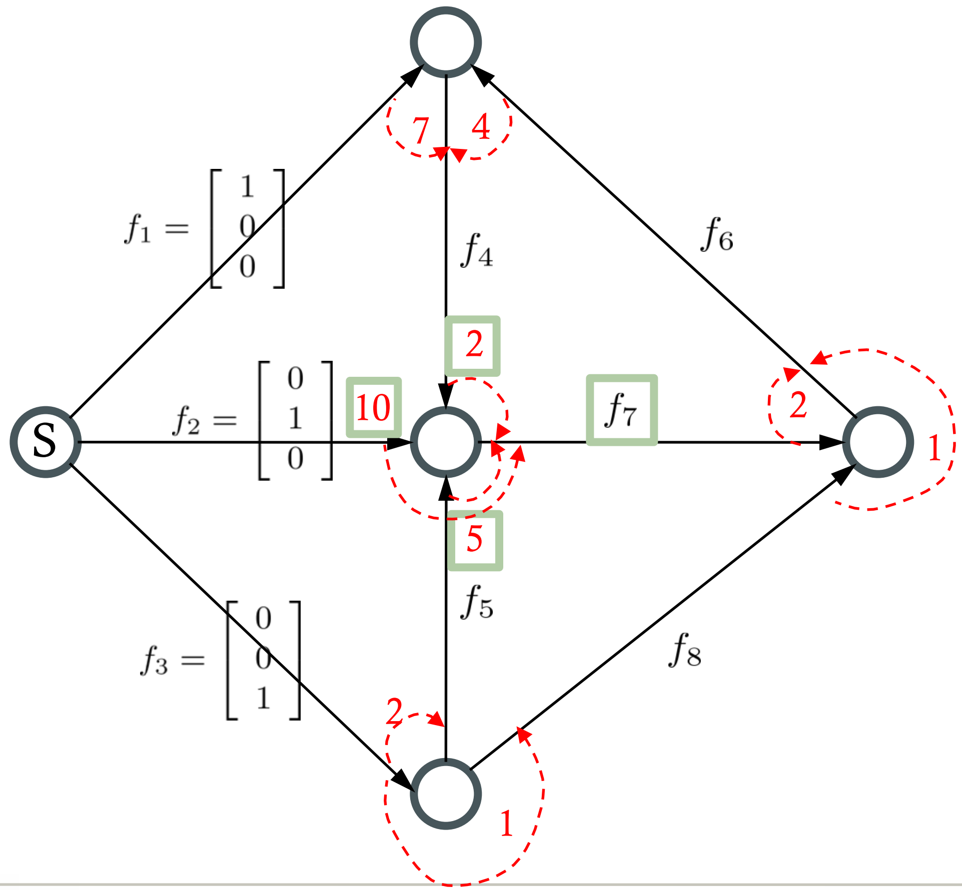
$$f_4 = 7 \cdot f_1 + 4 \cdot f_6$$
$$f_5 = 2 \cdot f_3$$
$$f_6 = 2 \cdot f_7 + 1 \cdot f_8$$
$$f_7 = 2 \cdot f_4 + 10 \cdot f_2 + 5 \cdot f_5$$
$$f_8 = 1 \cdot f_3$$

Random coefficients

Field size = 11

New Algebraic Formulation

(3) Require each vector to be a linear combination of its incoming vectors.



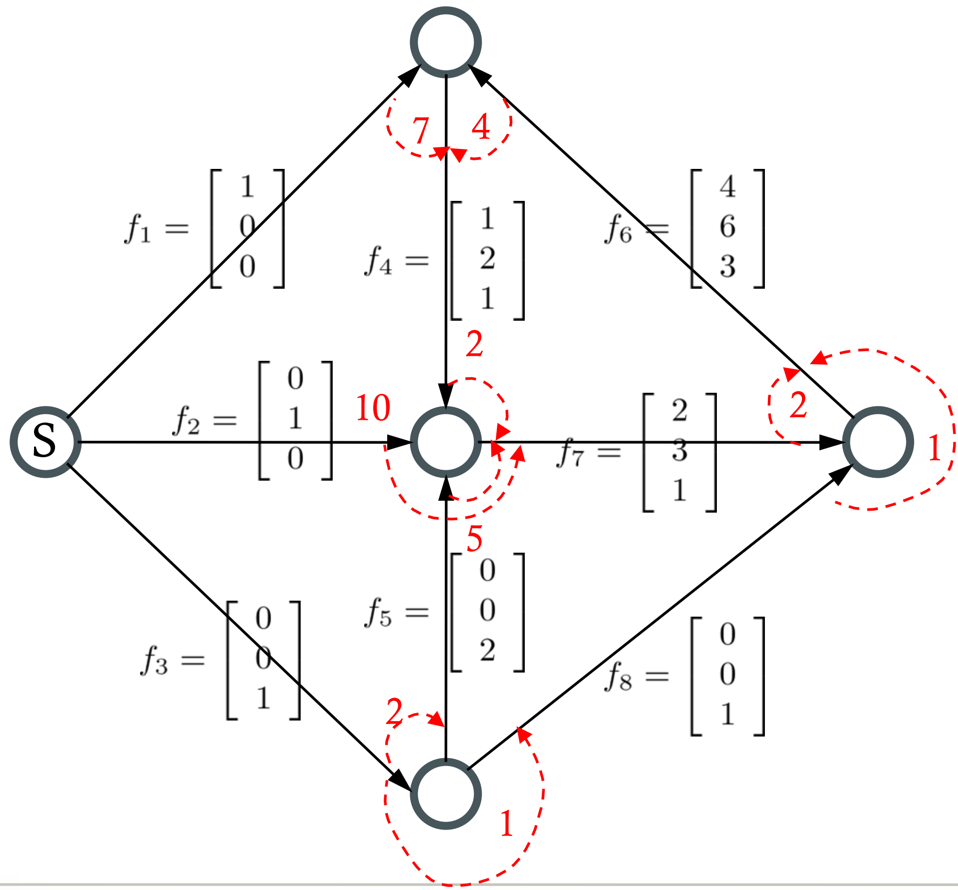
$$f_4 = 7 \cdot f_1 + 4 \cdot f_6$$
$$f_5 = 2 \cdot f_3$$
$$f_6 = 2 \cdot f_7 + 1 \cdot f_8$$
$$f_7 = 2 \cdot f_4 + 10 \cdot f_2 + 5 \cdot f_5$$
$$f_8 = 1 \cdot f_3$$

Random coefficients

Field size = 11

New Algebraic Formulation

(4) Compute vectors that satisfy all the equations.

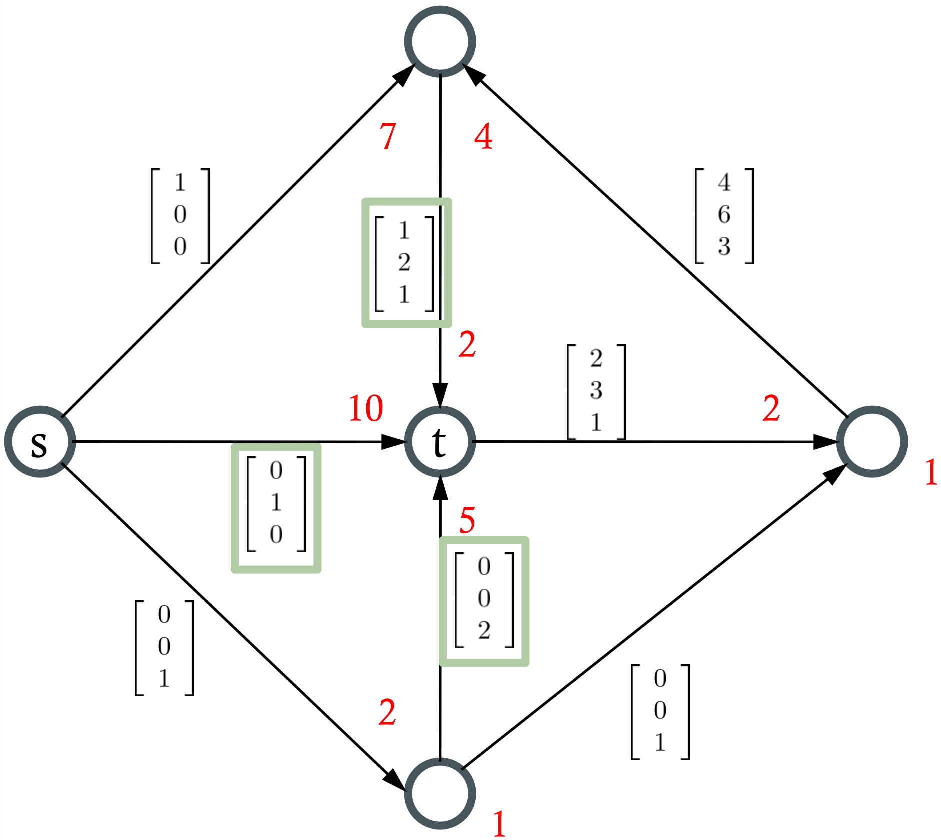


$$f_4 = 7 \cdot f_1 + 4 \cdot f_6$$
$$f_5 = 2 \cdot f_3$$
$$f_6 = 2 \cdot f_7 + 1 \cdot f_8$$
$$f_7 = 2 \cdot f_4 + 10 \cdot f_2 + 5 \cdot f_5$$
$$f_8 = 1 \cdot f_3$$

Random coefficients

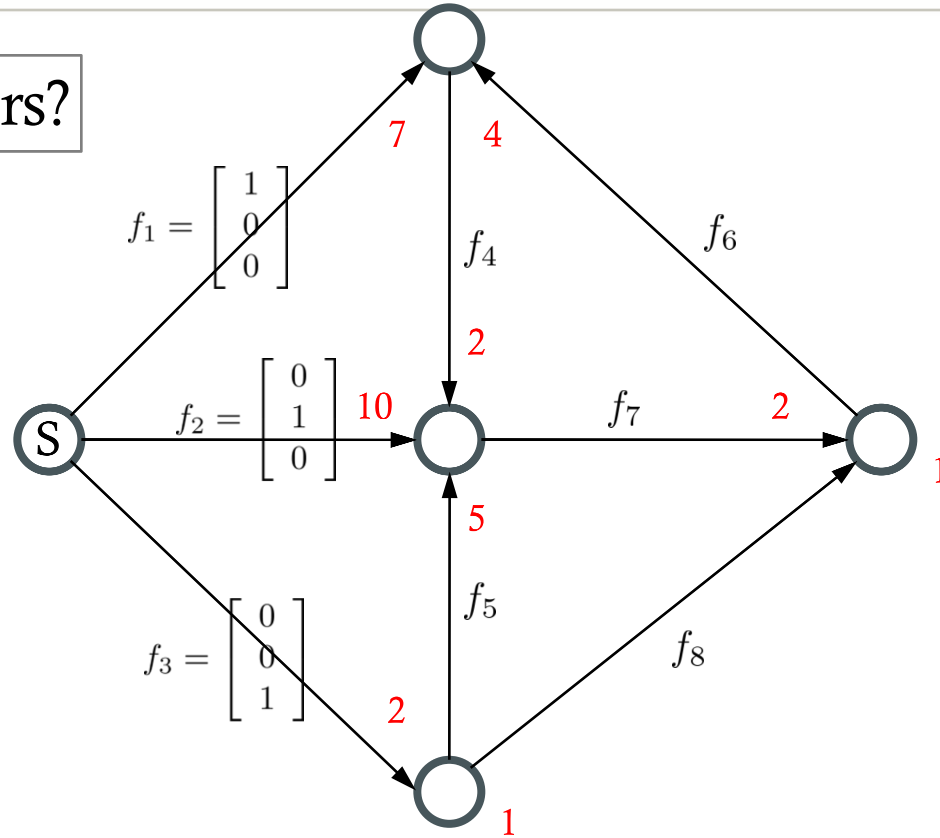
Field size = 11

Theorem: Field size is poly(m), with *high probability* for every vertex v, the rank of incoming vectors to v is equal to the edge connectivity from s to v



e.g. s-t connectivity = rank $\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 2 \end{pmatrix}$

How to compute those vectors?



$$f_4 = 7 \cdot f_1 + 4 \cdot f_6$$
$$f_5 = 2 \cdot f_3$$
$$f_6 = 2 \cdot f_7 + 1 \cdot f_8$$
$$f_7 = 2 \cdot f_4 + 10 \cdot f_2 + 5 \cdot f_5$$
$$f_8 = 1 \cdot f_3$$

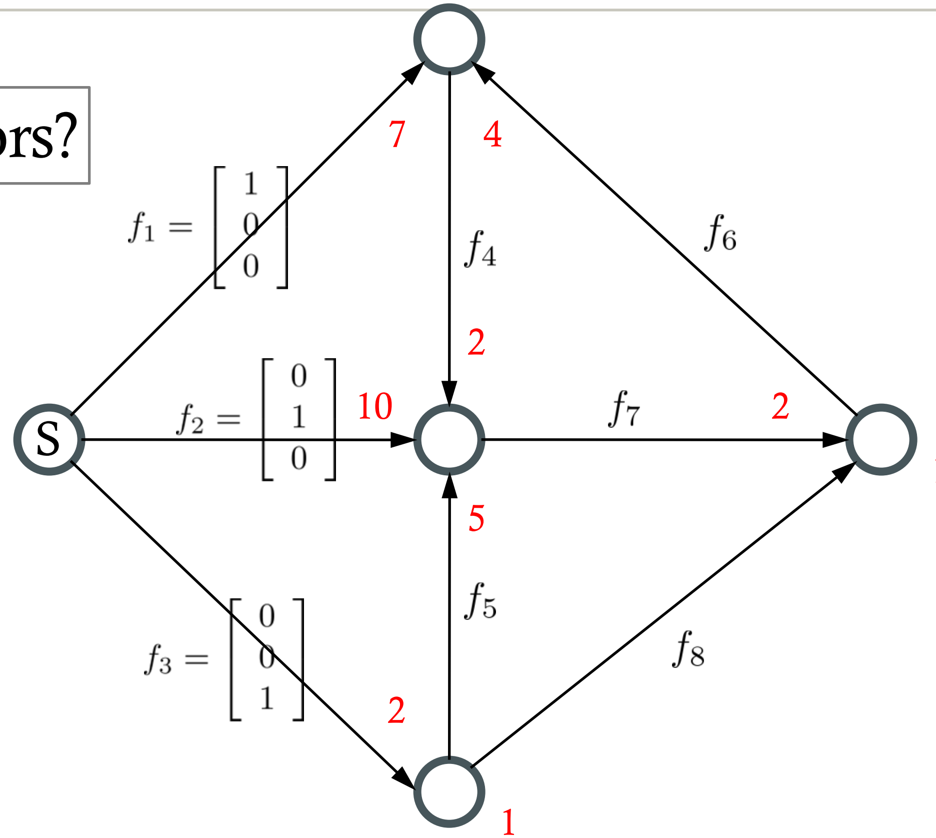
$$\begin{pmatrix} | & | & & | \\ f_1 & f_2 & \cdots & f_8 \\ | & | & & | \end{pmatrix} = \begin{pmatrix} | & | & & | \\ f_1 & f_2 & \cdots & f_8 \\ | & | & & | \end{pmatrix} \begin{pmatrix} & & & 7 & & & & \\ & & & & & & 10 & \\ & & & & 2 & & & 1 \\ & & & & & & 2 & \\ & & & & & & 5 & \\ & & & 4 & & & & \\ & & & & & 2 & & \\ & & & & & 1 & & \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & & & & & \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & & & & & \end{pmatrix}$$

$$F \quad = \quad F \quad K \quad + \quad H$$

How to compute those vectors?

$$F = FK + H$$

$$F = H(I - K)^{-1}$$



$$\begin{pmatrix} | & | & & | \\ f_1 & f_2 & \cdots & f_8 \\ | & | & & | \end{pmatrix} = \begin{pmatrix} | & | & & | \\ f_1 & f_2 & \cdots & f_8 \\ | & | & & | \end{pmatrix} \begin{pmatrix} & & & 7 & & & & \\ & & & & & & 10 & \\ & & & & 2 & & & 1 \\ & & & & & & 2 & \\ & & & & & & 5 & \\ & & & 4 & & & & \\ & & & & & 2 & & \\ & & & & & 1 & & \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & & & & & \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & & & & & \end{pmatrix}$$

$$F \quad = \quad F \quad\quad K \quad + \quad H$$

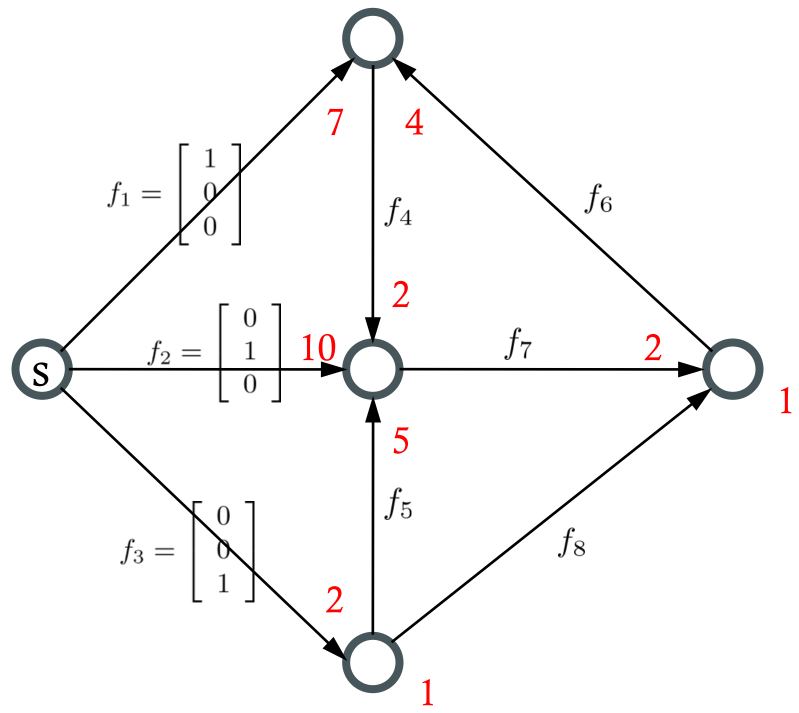
Algorithmic Results

- Advantages:
  - compute edge-connectivity from one source to all vertices at the same time
  - Allow use of fast algorithms from linear algebra

- Faster Algorithms
  - Single source / All pairs edge connectivities
  - General / Acyclic / Planar graphs

General Directed Graphs

$$F = H(I - K)^{-1}$$



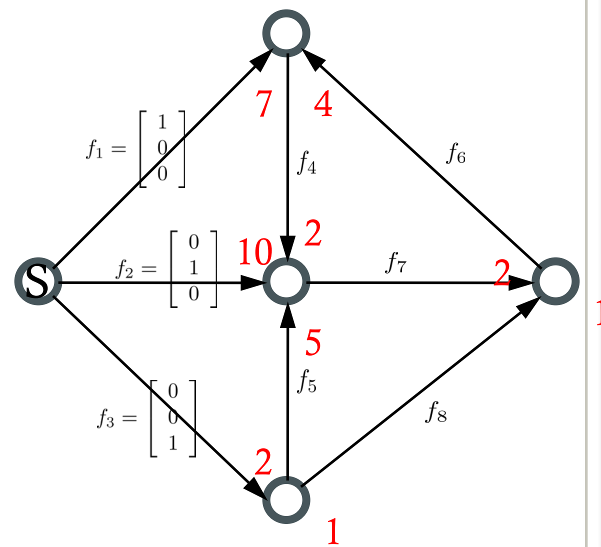
- S-v connectivity = rank of vectors going into v
- Computing F takes $O(m^\omega)$ time

Multiple Sources

$$F = H(I-K)^{-1}$$

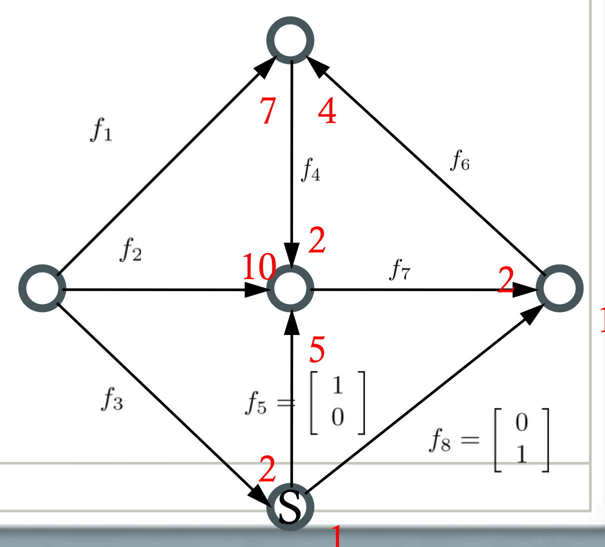
For source 1

$$\begin{pmatrix} | & | & & | \\ f_1 & f_2 & \cdots & f_8 \\ | & | & & | \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & & & & & \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & & & & & \end{pmatrix} \cdot \begin{pmatrix} 1 & & & -7 & & & & \\ & 1 & & & & & -10 & \\ & & 1 & & -2 & & & 1 \\ & & & 1 & & & -2 & \\ & & & & 1 & & -5 & \\ & & & -4 & & 1 & & \\ & & & & & -2 & 1 & \\ & & & & & -1 & & 1 \end{pmatrix}^{-1}$$

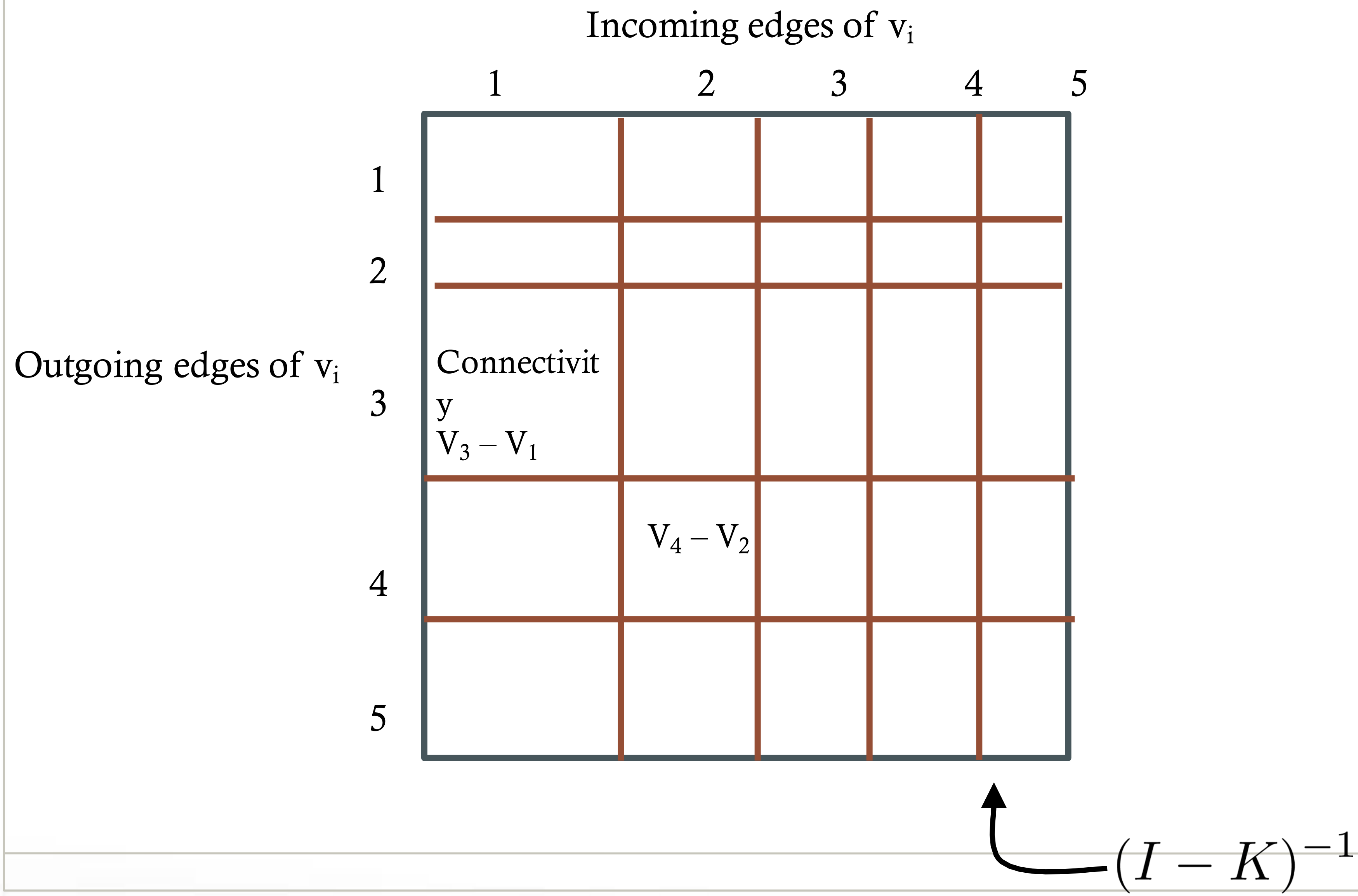


Another source

$$\begin{pmatrix} | & | & & | \\ f_1 & f_2 & \cdots & f_8 \\ | & | & & | \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & & & -7 & & & & \\ & 1 & & & & & -10 & \\ & & 1 & & -2 & & & 1 \\ & & & 1 & & & -2 & \\ & & & & 1 & & -5 & \\ & & & -4 & & 1 & & \\ & & & & & -2 & 1 & \\ & & & & & -1 & & 1 \end{pmatrix}^{-1}$$



All-Pairs Edge-Connectivities



All-Pairs Edge-Connectivity

Encoding: $O(m^w)$ (to compute the inverse)

Decoding: $O(m^2 n^{w-2})$ (to compute the rank of all submatrices)

Overall: $O(m^w)$

Best known (combinatorial) methods: $O(\min(n^{2.5}m, m^2n, n^2m^{10/7}))$

Sparse graphs: $m=O(n)$, algebraic algorithm takes $O(n^w)$ steps while other algorithms takes $O(n^3)$ steps.

New Questions

- Is there some combinatorial structure that the algebraic structure is exploiting that we have not found yet?
- Can we obtain algorithms without using fast matrix multiplication?
- Does the algebraic methodology work for other connectivity problems?

Benefits *to* Combinatorial Optimization

My perspective/experience

- New applications of existing results
- New problems
- New algorithms for classical problems
- Challenging open problems
- Interdisciplinary collaborations/friendships

Personal Benefits

- Collaborations with ECE/Info theory. Christina Fragouli, Emina Soljanin, Serap Savari, Pramod Viswanath, Sreeram Kannan, Adnan Raja, Sudeep Kamath …
- Conversations with several CS researchers on interrelated topics
- Several papers. Direct and indirect!
- Made me understand my own area better!
- Friendships and fun

Thanks!