On Multi-source Multi-sink Hyperedge Networks: Enumeration, Rate Region Computation, & Hierarchy

Congduan Li

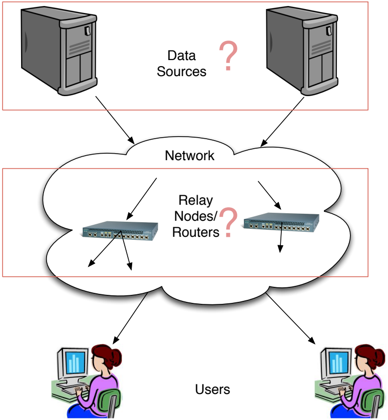
congduan.li@gmail.com

Joint work with Steven Weber and John Walsh

Jun 15, 2016

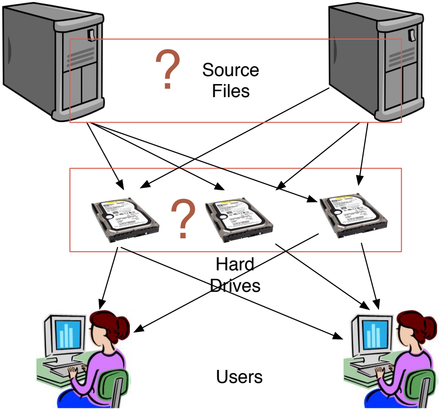
Motivation: information flow in networks

- Coding (mix the input) outperforms routing (receive-forward) in many circumstances
- Efficient information flow in networks: admissible source rate v.s channel capacities



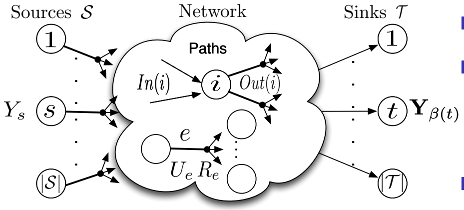
Motivation: information storage in data centers

- Coding (mix sources) saves resources, compared with simple replication
- Efficient information storage in data centers: admissible source file size v.s hard disk capacities



Generalized model

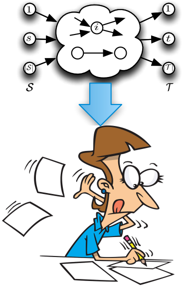
A multisource multisink hyperedge network (hyperedge MSNC)



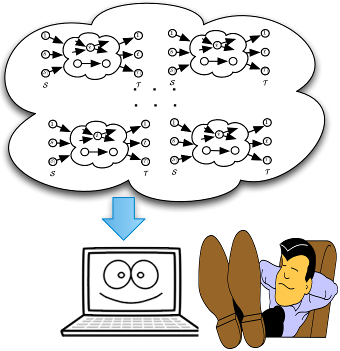
- Rate/capacity region: compute
- Notation: $(K, |\mathcal{E}_U|)$ means K sources, $|\mathcal{E}_U|$ intermediate hyperedges
- We made three major contributions

Contribution 1: Revolution, use computer for rate region calculation

- Conventional: 1 (special) network, info. ineq., manually, 1 paper
- Computational: $10^3, 10^6, 10^{12}$ (arbitrary) networks, computer, # of papers?



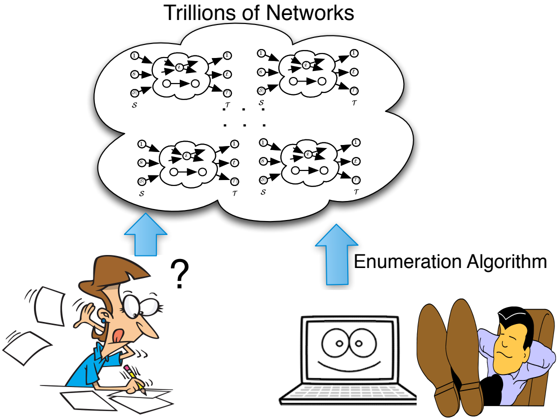
Conventional way



Computational way

Contribution 2: enumeration of all networks

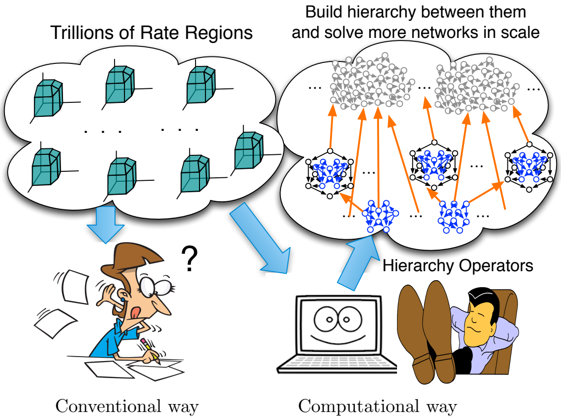
- Conventional: almost impossible to list all networks due to the large number of instances
- Computational: enumerate $\geq 10^9, 10^{12}$ general networks



Conventional way   Computational way

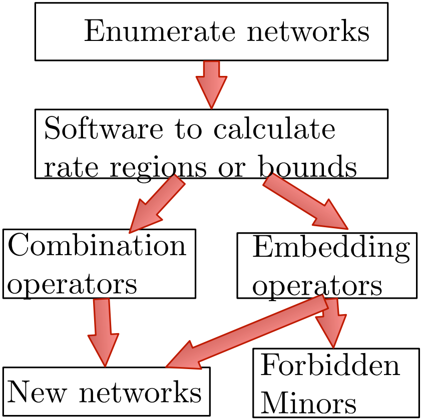
Contribution 3: build hierarchy between networks

- Conventional: what to do with so many rate regions?
- Computational: define embedding and combination operators, build a hierarchy, analyze the rate regions and use them to solve even more networks in scale



Outline

1. Enumeration
   - Network Model
   - Minimality
   - Network Equivalence Class
   - Enumeration algorithm
2. Rate Region Computation
3. Network Operations
   - Embedding Operations
   - Combination Operations
   - Play with both

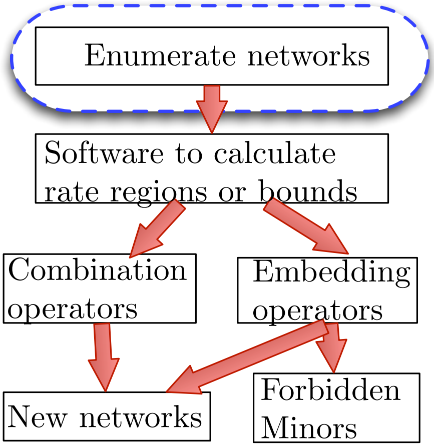


Outline

1 Enumeration
- Network Model
- Minimality
- Network Equivalence Class
- Enumeration algorithm

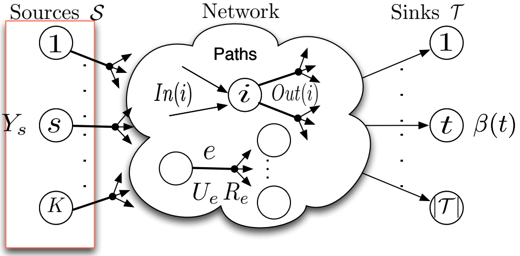
2 Rate Region Computation

3 Network Operations
- Embedding Operations
- Combination Operations
- Play with both



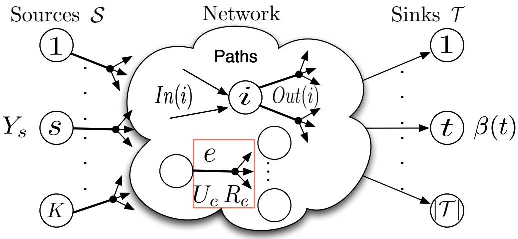
Network Coding: Sources

We assume $U_{\mathrm{Out}(s)} = Y_s$



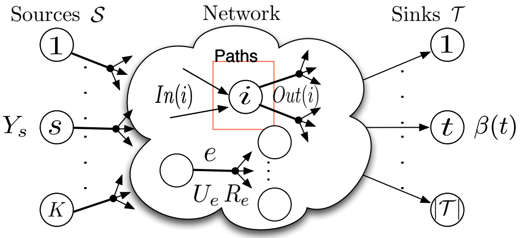
Source Rate	$H(Y_s) \geq \omega_s$
Source Independence	$H(Y_{\mathcal{S}}) = \sum_{s\in\mathcal{S}} H(Y_s)$
Source Encoding	$H(U_{\mathrm{Out}(s)} \| Y_s) = 0$

Network Coding: Edges

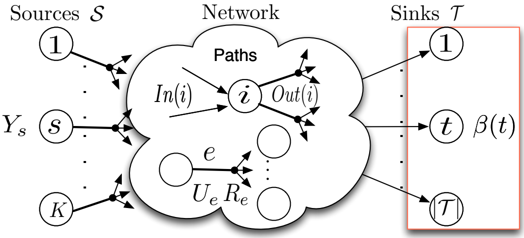


Coding Rate $$R_e \geq H(U_e), \; e \in \mathcal{E}$$

Network Coding: Nodes

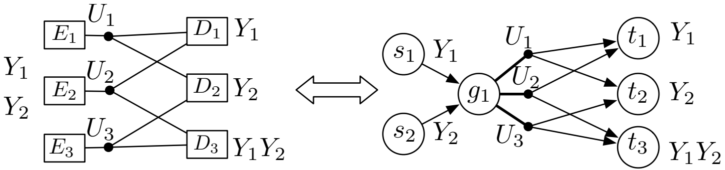


Coding Constraints $\quad H(U_{\mathrm{Out}(i)}|U_{\mathrm{In}(i)}) = 0, \ i \in \mathcal{V} \backslash (\mathcal{S} \cup \mathcal{T})$



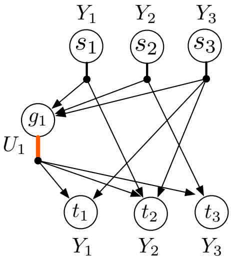
Decoding Constraints $$H(Y_{\beta(t)}|U_{\mathrm{In}(t)}) = 0,\ t \in \mathcal{T}$$

Special Case: Independent Distributed Source Coding (IDSC)



- Sources available to all encoders
- Decoders demand various subsets of of sources
- When sources are prioritized, it becomes MDCS

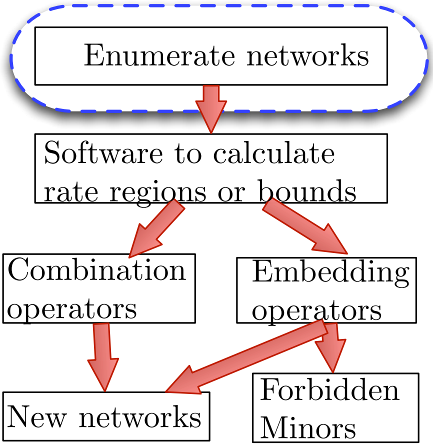
Special Case: Index Coding



- Only one intermediate edge that transmits all information of sources
- Sources may be directly available at sinks as side information
- $(K, 1)$ hyperedge MSNC

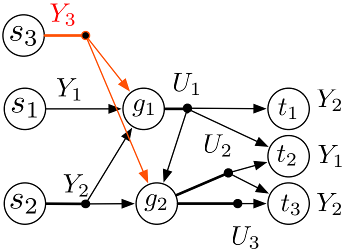
Outline

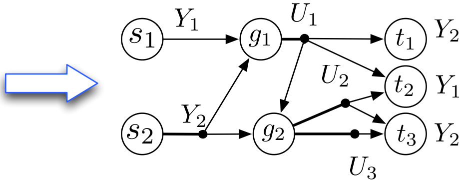
1. Enumeration
   - Network Model
   - Minimality
   - Network Equivalence Class
   - Enumeration algorithm
2. Rate Region Computation
3. Network Operations
   - Embedding Operations
   - Combination Operations
   - Play with both



Source minimality example

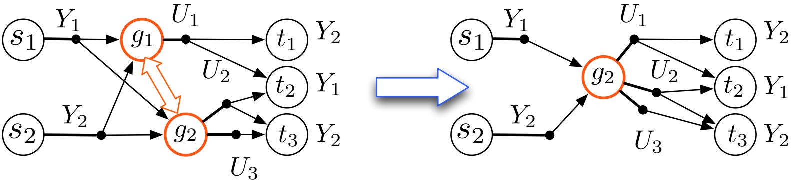
Source minimality example: Redundant source s_3 is not demanded by any sink.





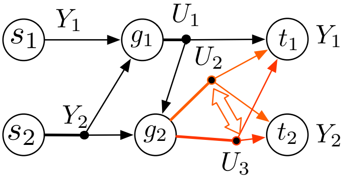
Node minimality example

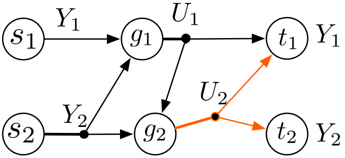
Node minimality example: g_1, g_2 can be merged due to same input



Edge minimality example

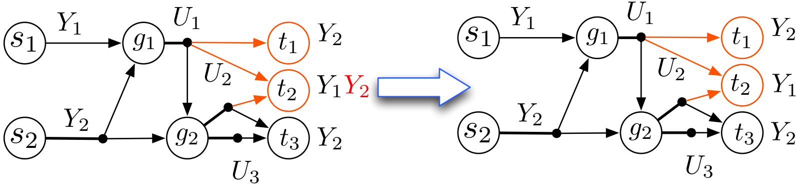
Edge minimality example: U_2, U_3 are parallel and can be merged





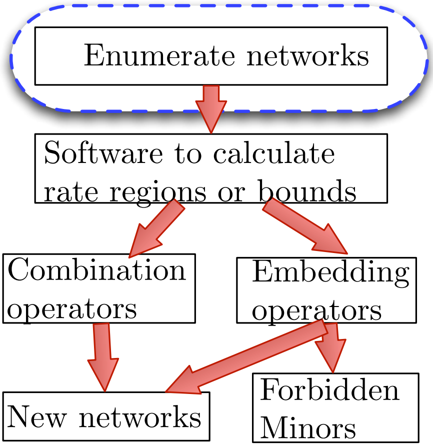
Sink minimality example

Sink minimality example: decoding ability of Y_2 at t_2 implied by t_1, equivalent to let t_2 demand Y_1 only

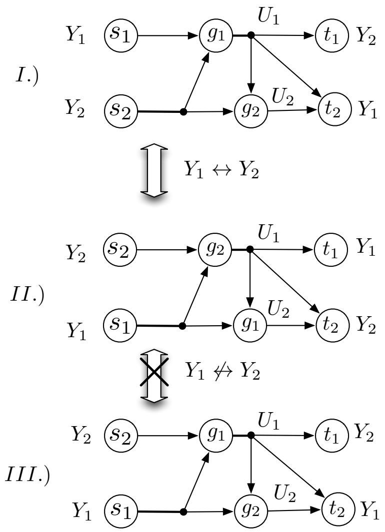


Outline

1. Enumeration
   - Network Model
   - Minimality
   - Network Equivalence Class
   - Enumeration algorithm
2. Rate Region Computation
3. Network Operations
   - Embedding Operations
   - Combination Operations
   - Play with both



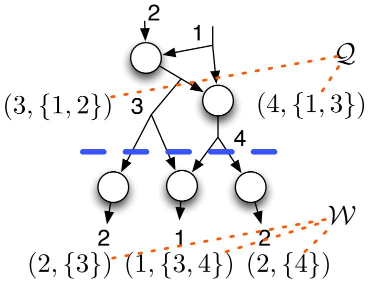
Equivalent Networks



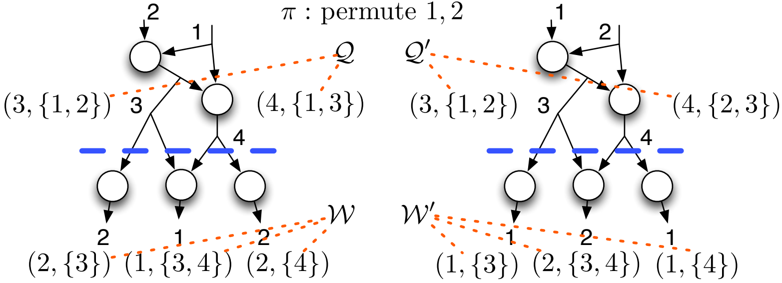
- *I*, *II* are equivalent: permute sources and edges
- *I*, *III* are not equivalent: sources are only permuted at source side
- *II*, *III* are not equivalent: sources are only permuted at sink side

Representing a Network

- Ordered pair $(\mathcal{Q}, \mathcal{W})$, sources $1 \dots K$, edges $K+1, \dots, K+|\mathcal{E}_U|$;
- Edge definitions $\mathcal{Q} \subseteq \{(i, \mathcal{A}) | i \in \{K+1, \dots, K+|\mathcal{E}_U|\},\ \mathcal{A} \subseteq \{1, \dots, K+|\mathcal{E}_U|\} \setminus \{i\}\}$;
- Sink definitions $\mathcal{W} \subseteq \{(i, \mathcal{A}) | i \in \{1, \dots, K\},\ \mathcal{A} \subseteq \{1, \dots, K+|\mathcal{E}_U|\} \setminus \{i\}\}$;
- Same i allowed to appear in $\mathcal{W}$ but not in $\mathcal{Q}$

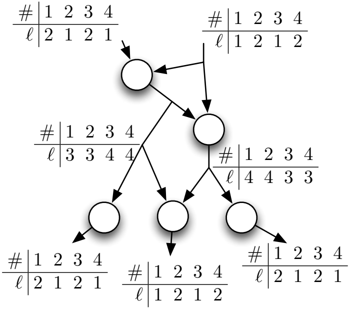


- Symmetry group $\mathbf{G} := S_{\{1,2,\ldots,K\}} \times S_{\{K+1,\ldots,K+|\mathcal{E}_U|\}}$;
- $\pi \in \mathbf{G}$, then $\pi(\mathcal{Q}) \mapsto \{\pi((i,\mathcal{A}))|(i,\mathcal{A}) \in \mathcal{Q}\}$;
- $\pi((\mathcal{Q},\mathcal{W})) = (\pi(\mathcal{Q}),\pi(\mathcal{W}))$;
- Isomorphic or Equivalent: $\exists \pi \in \mathbf{G}$ such that $\pi((\mathcal{Q}_1,\mathcal{W}_1)) = (\mathcal{Q}_2,\mathcal{W}_2)$.



Canonical Network: minimal representative in each orbit

- Orbit: $\mathcal{O}_{(\mathcal{Q},\mathcal{W})} := \{(\pi(\mathcal{Q}),\pi(\mathcal{W}))|\pi \in \mathbf{G}\}$
- Networks in an orbit are isomorphic or equivalent to each other
- Transversal: one representative for each orbit, the canonical one
- Lexicographically order the pairs $(i,\mathcal{A})$ according to $(i,\mathcal{A}) > (j,\mathcal{A}')$ if $j < i$ or $i = j, \mathcal{A}' < \mathcal{A}$ under the lexicographic ordering
- Canonical: apply order to $(\mathcal{Q},\mathcal{W})$ and get the minimal one

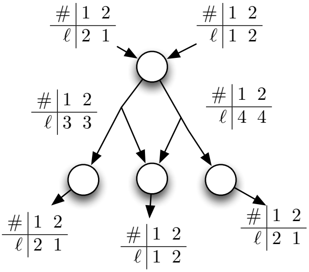


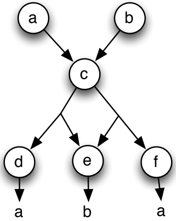
1 is the canonical representative

#	$\mathcal{Q}$	$\mathcal{W}$
1	$\{(3,\{1,2\}),(4,\{1,3\})\}$	$\{(1,\{3,4\}),(2,\{3\}),(2,\{4\})\}$
2	$\{(3,\{1,2\}),(4,\{2,3\})\}$	$\{(1,\{3\}),(1,\{4\}),(2,\{3,4\})\}$
3	$\{(3,\{1,4\}),(4,\{1,2\})\}$	$\{(1,\{3,4\}),(2,\{3\}),(2,\{4\})\}$
4	$\{(3,\{2,4\}),(4,\{1,2\})\}$	$\{(1,\{3\}),(1,\{4\}),(2,\{3,4\})\}$

Why this representation

- Some minimality constraints easily to take care, e.g., repeated edges, redundant nodes, etc
- Smaller orbits, compared with node representation
- Orbit-stabilizer theorem: $|\mathcal{O}_{(\mathcal{Q},\mathcal{W})}| = \frac{|G|}{|Stab((\mathcal{Q},\mathcal{W}))|}$





#	a	b	c	d	e	f
1	2	1	3	4	5	6
2	1	2	3	4	5	6
3	2	1	4	5	6	3
⋮						
⋮						
24	1	2	3	4	6	5

Subgroup of $\mathcal{S}_{\{a,b\}} \times \mathcal{S}_{\{c,d,e,f\}}$ stabilizing $(\mathcal{V},\mathcal{E}) = \mathcal{S}_{\{d,f\}}$

Isomorphs in Node Representation (24)

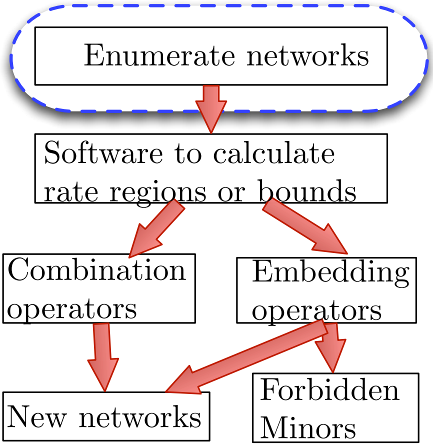
#	$\mathcal{Q}$	$\mathcal{W}$
1	$\{(3,\{1,2\}),(4,\{1,2\})\}$	$\{(1,\{3,4\}),(2,\{3\}),(2,\{4\})\}$
2	$\{(3,\{1,2\}),(4,\{1,2\})\}$	$\{(1,\{3\}),(1,\{4\}),(2,\{3,4\})\}$

Isomorphs in Edge and Sink Definition Representation (2)

Subgroup of $\mathcal{S}_{\{1,2\}} \times \mathcal{S}_{\{3,4\}}$ stabilizing $(\mathcal{G},\mathcal{T}) = \mathcal{S}_{\{3,4\}}$

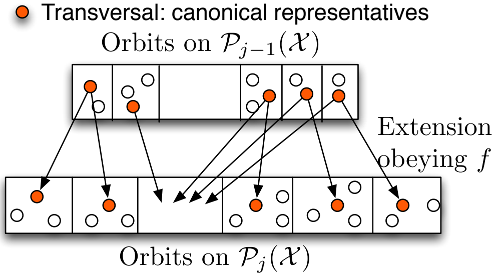
Outline

1. Enumeration
   - Network Model
   - Minimality
   - Network Equivalence Class
   - Enumeration algorithm
2. Rate Region Computation
3. Network Operations
   - Embedding Operations
   - Combination Operations
   - Play with both



Leiterspiel Algorithm [BettenBraunFripertinger, 2006]

- Computes transversal of orbits on size j subsets $\mathcal{P}_j(\mathcal{X})$ of set $\mathcal{X}$, incrementally in j; also gives symmetry group (stabilizer)
- Lists directly the canonical representatives satisfying some test function f, as long as this test has the inherited property, i.e., if a superset satisfies, its subsets also satisfy
- Input: set $\mathcal{X}$, group G, inherited test function f
- Output: transversal of subsets of different sizes until stop, either fixed j or other stop conditions, symmetry group (stabilizer)



Enumeration based on Leiterspiel Algorithm

- Target: $(K, |\mathcal{E}_U|)$ non-isomorphic networks
- Recall representation of networks: $(\mathcal{Q}, \mathcal{W})$, list pool for $\mathcal{Q}$ first
- Leiterspiel defines edges incrementally from 1 to $|\mathcal{E}_U|$

$$\boxed{\mathcal{X} := \{(i, \mathcal{A})\,|i \in \{K+1, \dots, K+|\mathcal{E}_U|\},\ \mathcal{A} \subseteq \{1, \dots, K+|\mathcal{E}_U|\} \setminus \{i\}\}}$$

$$\boxed{\text{Acting group } G := S_{\{1,\dots,K\}} \times S_{\{K+1,\dots,K+|\mathcal{E}_U|\}}}$$

With some minimality constraints inherited

$$\boxed{\text{Call Leiterspiel } T_{|\mathcal{E}_U|} = Leiterspiel(G, \mathcal{P}^f_{|\mathcal{E}_U|}(\mathcal{X}))}$$

Stop condition: reach number of edges

Enumeration based on Leiterspiel Algorithm

- Now for each $\mathcal{Q} \in T_{|\mathcal{E}_U|}$, list possible $\mathcal{W}$
- Leiterspiel incrementally adds sinks from 1 to no possible new sink

$$\mathcal{Y} := \{(i, \mathcal{A}) \,|\, \exists \text{ a directed path in } \mathcal{Q} \text{ from } i \text{ to at least one edge in } \mathcal{A}\}$$

$$\text{Acting group } G := S_{\{1,\ldots,K\}} \times S_{\{K+1,\ldots,K+|\mathcal{E}_U|\}}$$

With some minimality constraints inherited

$$\text{Call Leiterspiel } T_{|\mathcal{T}|} = Leiterspiel(G, \mathcal{P}^f_{|\mathcal{T}|}(\mathcal{Y}))$$

Stop condition: cannot increase j obeying f

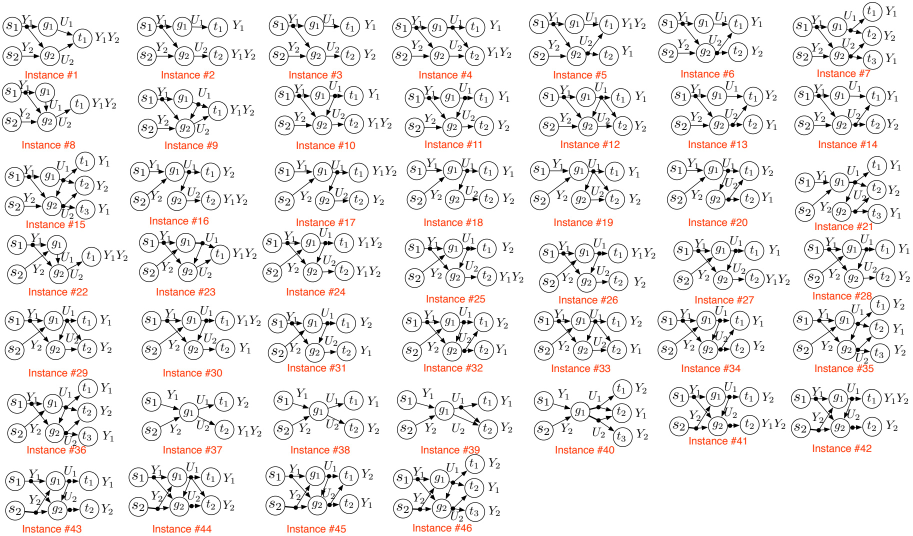
For each $\mathcal{W} \in \{T_1, \ldots, T_{|\mathcal{T}|}\}$, test all the other minimality conditions on $(\mathcal{Q}, \mathcal{W})$, if it passes, we obtain a non-isomorphic network instance

Enumeration Results

$|\mathcal{M}|$: number of non-isomorphic networks, listed;
$|\hat{\mathcal{M}}|$: number of networks with edge isomorphism, counted;
$|\hat{\mathcal{M}}_n|$: number of networks with node isomorphism, counted;

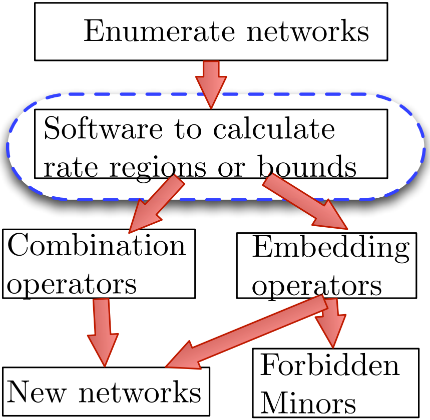
$(K, \lvert\mathcal{E}_U\rvert)$	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)
$\lvert\mathcal{M}\rvert$	4	132	1	333	485 890	9	239 187
$\lvert\hat{\mathcal{M}}\rvert$	7	749	1	1 270	5 787 074	31	2 829 932
$\lvert\hat{\mathcal{M}}_n\rvert$	39	18,401	6	$\geq 10^5$	$\geq 10^{12}$	582	$\geq \times 10^{11}$

All $(2, 2)$ networks: no direct access btw sources & sinks



Outline

1. Enumeration
   - Network Model
   - Minimality
   - Network Equivalence Class
   - Enumeration algorithm
2. **Rate Region Computation**
3. Network Operations
   - Embedding Operations
   - Combination Operations
   - Play with both



Rate region

- Rate region: all possible rate and source entropy vectors satisfying all network constraints.
- Collect the N network random variables and their joint entropies.
- Define Γ_N^*: $2^N - 1$-dim., region of valid entropy vectors. (revisit later)
- Constraints from network A:

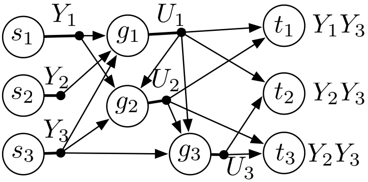
$$\begin{aligned}
\mathcal{L}_1 &= \{\mathbf{h} \in \Gamma_N^* : h_{Y_\mathcal{S}} = \Sigma_{s\in\mathcal{S}} h_{Y_s}\} && (1)\\
\mathcal{L}_2 &= \{\mathbf{h} \in \Gamma_N^* : h_{X_{\mathrm{Out}(k)}|Y_s} = 0\} && (2)\\
\mathcal{L}_3 &= \{\mathbf{h} \in \Gamma_N^* : h_{X_{\mathrm{Out}(i)}|X_{\mathrm{In}(i)}} = 0\} && (3)\\
\mathcal{L}_4 &= \{(\mathbf{h}^T, \mathbf{R}^T)^T \in \mathbb{R}_+^{2^N-1+|\mathcal{E}|} : R_e \geq h_{U_e}, e \in \mathcal{E}\} && (4)\\
\mathcal{L}_5 &= \{\mathbf{h} \in \Gamma_N^* : h_{Y_{\beta(t)}|U_{\mathrm{In}(t)}} = 0\}. && (5)
\end{aligned}$$

- Rate region (cone) in terms of edge rates and source entropies (derived from [Yan, Yeung, Zhang TranIT 2012]):

$$\mathcal{R}_*(\mathrm{A}) = \mathrm{proj}_{R_\mathcal{E}, H(Y_\mathcal{S})}(\overline{\mathrm{con}(\Gamma_{\mathrm{N}}^* \cap \mathcal{L}_{123})} \cap \mathcal{L}_{45}) \qquad (6)$$

Rate Region Example

- A $(3,3)$ network and its rate region $\mathcal{R}_*(\mathrm{A})$
- Rate region: a cone with dimensions of all variables in the network



$$
\begin{aligned}
R_1 &\geq H(Y_2)\\
R_1+R_3 &\geq H(Y_2)+H(Y_3)\\
R_2+R_3 &\geq H(Y_2)+H(Y_3)\\
R_1+R_2 &\geq H(Y_1)+H(Y_2)+H(Y_3)\\
R_1+R_2+2R_3 &\geq H(Y_1)+2H(Y_2)+2H(Y_3)
\end{aligned}
$$

Rate region

- Rate region: a cone in terms of edge rates and source entropies:

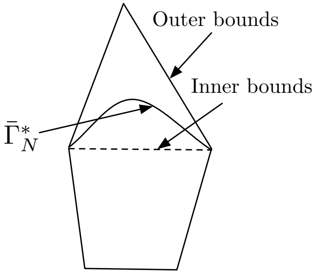
$$\mathcal{R}_*(\mathsf{A}) = \mathrm{proj}_{R_{\mathcal{E}}, H(Y_{\mathcal{S}})}(\overline{\mathrm{con}(\Gamma^*_{\mathrm{N}} \cap \mathcal{L}_{123})} \cap \mathcal{L}_{45}) \tag{7}$$

- Involves Γ^*_N: $2^N - 1$-dim., region of valid entropy vectors.

Region of Entropic Vectors

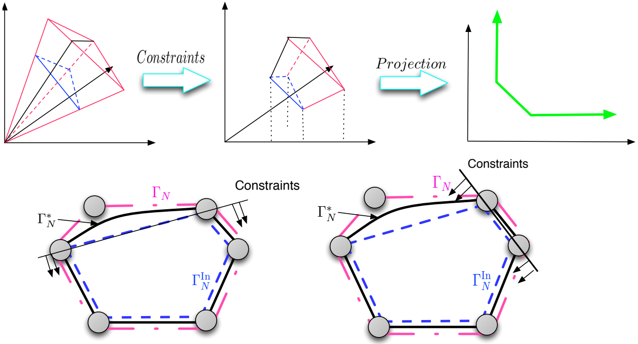
Γ_N^*:

- Open in general
- $\bar{\Gamma}_N^*$ not fully characterized for $N \geq 4$: convex but contains non-polyhedral part



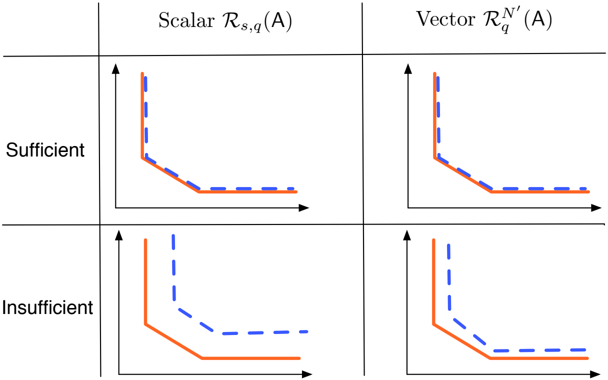
Sandwich Bounds

- $\bar{\Gamma}^*_N \to \Gamma^{Out}_N$: $\mathcal{R}_{out}(A) = \mathrm{proj}_{R_{\mathcal{E}}, H(Y_{\mathcal{S}})}(\Gamma^{Out}_N \cap \mathcal{L}_{12345})$
- $\bar{\Gamma}^*_N \to \Gamma^{In}_N$: $\mathcal{R}_{in}(A) = \mathrm{proj}_{R_{\mathcal{E}}, H(Y_{\mathcal{S}})}(\Gamma^{In}_N \cap \mathcal{L}_{12345})$
- $\mathcal{R}_*(A) = \mathcal{R}_{out}(A) = \mathcal{R}_{in}(A)$, if $\mathcal{R}_{out}(A) = \mathcal{R}_{in}(A)$
- It becomes: Initial polyhedra $\to \cap$ constraints $\to$ projections
- Our work following this idea: Li, *et. al*, Allerton 2012, NetCod 2013, submission TransIT 2014.



Notion of sufficiency

- Outer bound typically used is Shannon outer bound→ $\mathcal{R}_o(\mathsf{A})$; inner bounds from representable matroids: scalar and vector bounds.
- Scalar sufficiency: $\boxed{\mathcal{R}_*(\mathsf{A}) = \mathcal{R}_{s,q}(\mathsf{A})}$
- Vector sufficiency: $\boxed{\mathcal{R}_*(\mathsf{A}) = \mathcal{R}_q^{N'}(\mathsf{A})}$



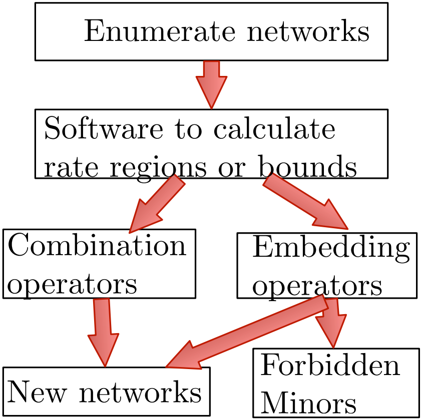
Rate Region Computation Results

$|\mathcal{M}|$: number of all network instances;
The other numbers represent the numbers of instances we can close the gap using various bounds, and hence exact rate region can be obtained

$(K, \|\mathcal{E}\|)$	$\|\mathcal{M}\|$	$\mathcal{R}_{s,2}(\mathrm{A})$	$\mathcal{R}_2^{N+1}(\mathrm{A})$	$\mathcal{R}_2^{N+2}(\mathrm{A})$	$\mathcal{R}_2^{N+4}(\mathrm{A})$
$(1, 2)$	4	4	4	4	4
$(1, 3)$	132	122	132	132	132
$(2, 1)$	1	1	1	1	1
$(2, 2)$	333	301	319	323	333
$(2, 3)$	485890	341406	403883	432872	–
$(3, 1)$	9	4	4	9	9
$(3, 2)$	239187	118133	168761	202130	–

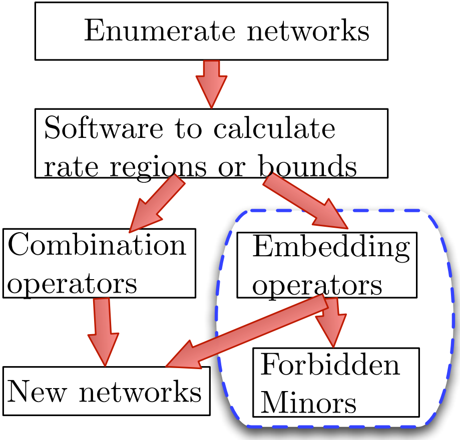
Outline

1. Enumeration
   - Network Model
   - Minimality
   - Network Equivalence Class
   - Enumeration algorithm
2. Rate Region Computation
3. Network Operations
   - Embedding Operations
   - Combination Operations
   - Play with both



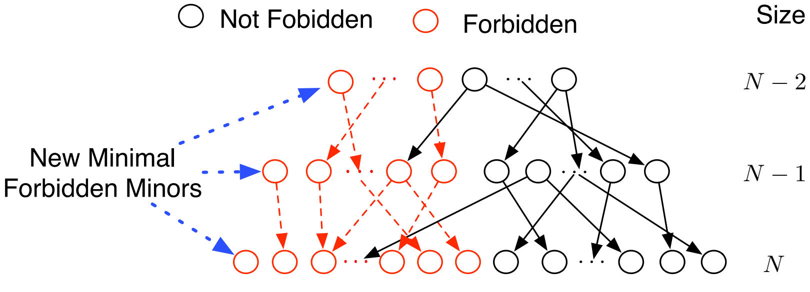
Outline

1. Enumeration
   - Network Model
   - Minimality
   - Network Equivalence Class
   - Enumeration algorithm
2. Rate Region Computation
3. Network Operations
   - Embedding Operations
   - Combination Operations
   - Play with both



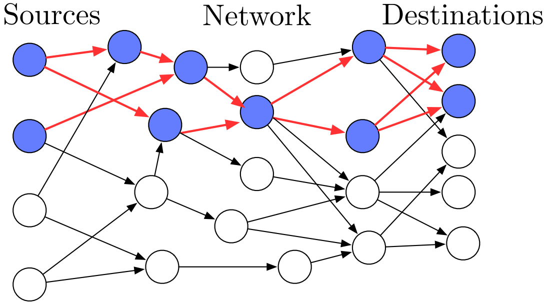
Motivated from graph and matroid theory

- Inheritance property regarding sufficiency of class of codes
- Minor-closed graphs: finite number of forbidden minors [RobertsonSeymour1983-2004]
- Rota's conjecture in matroid theory: finite number of forbidden minors for $\mathbb{F}_q$ representability [Oxley2011]



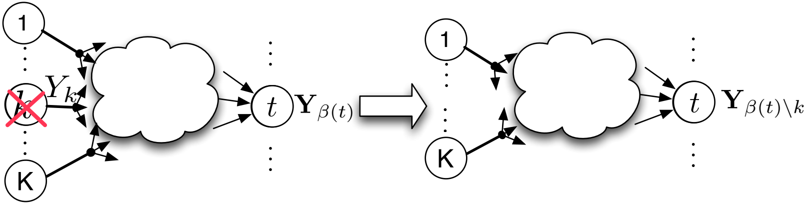
Similar characterization for networks?

- If networks have similar characterization? Possible list of forbidden embedded networks for sufficiency of linear codes over a field.
- Network operations to obtain such embedded networks preserving insufficiency, & region relationships
- Three operations



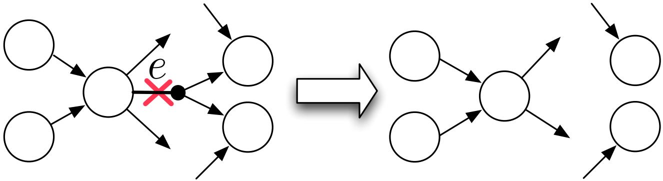
Source deletion

- $\mathrm{A}' = \mathrm{A} \setminus k$
- Source Y_k deleted, source k stops sending information to the network, $H(Y_k) = 0$
- Sinks requiring Y_k will no longer demand it.



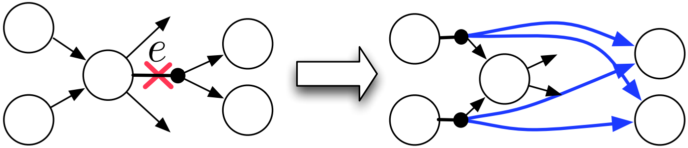
Edge deletion

- $A' = A \setminus e$
- Edge e deleted, nothing on U_e, $R_e = H(U_e) = 0$.



Edge contraction

- $\mathsf{A}' = \mathsf{A}/e$
- Edge e contracted, input to tail of e available for head of e, $R_e = \infty$, $H(U_e)$ free.

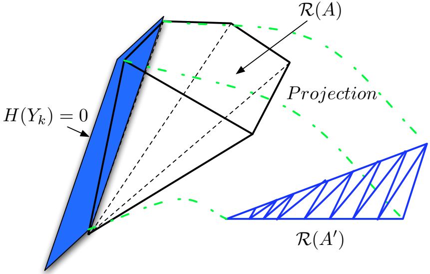


Source deletion: $\mathsf{A}' = \mathsf{A} \setminus k$

For each $i \in \{*, q, (s,q), o\}$,

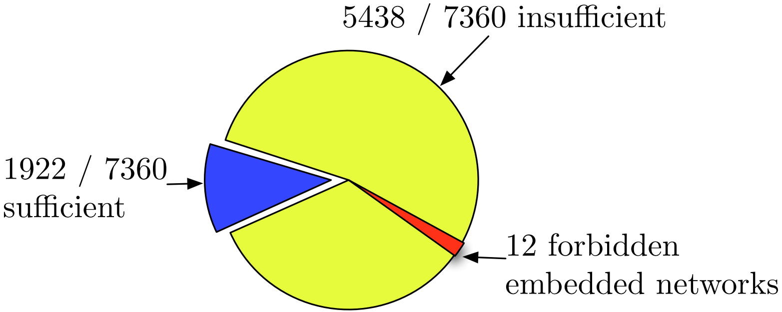
$$\mathcal{R}_i(\mathsf{A}') = \mathrm{Proj}_{\mathbf{Y}_{\setminus k}, \mathbf{R}_{\mathcal{E}'}} \left(\{ \mathbf{R} \in \mathcal{R}_i(\mathsf{A}) \,|\, H(Y_k) = 0 \} \right) \tag{8}$$

Sufficiency preserved from large to small network as the equation shows. Equivalently, insufficiency preserved from small to large network.



Example: Forbidden embedded networks

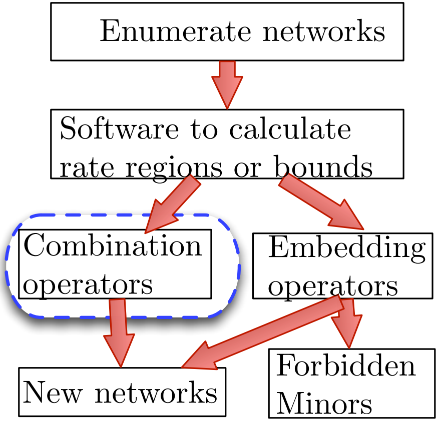
- Goal: minimal forbidden networks for sufficiency
- Scalar binary codes considered
- $k = 1, 2, 3; |\mathcal{E}| = 2, 3, 4$, 7360 non-isomorphic MDCS
- 1922 sufficient, 5438 insufficient
- 12 minimal forbidden minors (Li, *et. al* submission TransIT 2014)



Summary for embedding operators

- Three embedding operators: source deletion, edge deletion, edge contraction
- Rate region of smaller network derivable from the associated larger network
- Sufficiency of linear codes preserved from larger to smaller networks under embedding operations
- Equivalently, insufficiency of linear codes preserved from smaller networks to larger one

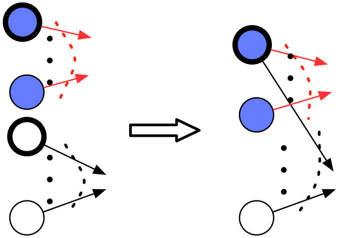
Outline



Enumerate networks
Software to calculate rate regions or bounds
Combination operators
Embedding operators
New networks
Forbidden Minors

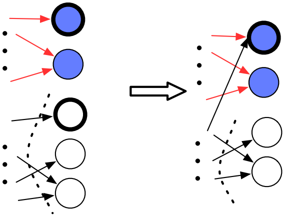
Source merge

- After merge: Merged sources serve as common sources to the two networks



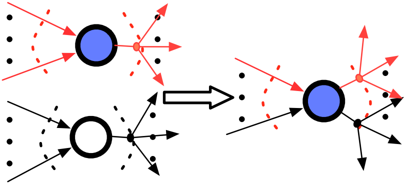
Sink merge

- After merge: Union the input and requests of sinks being merged, respectively



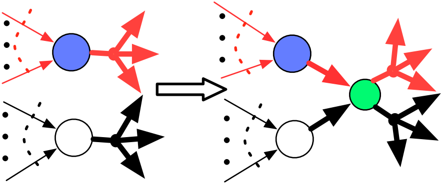
Intermediate node merge

- After merge: union input and output of the two nodes



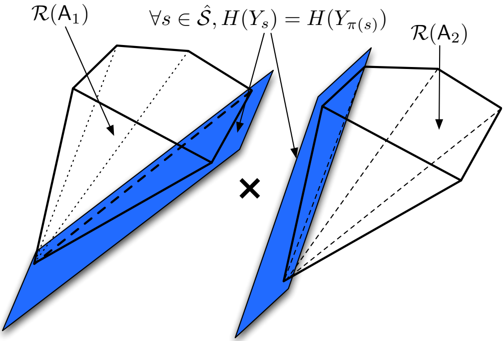
Edge merge

- Edge merge: create one extra node and four associated edges to replace the two original edges
- First two edges are ordinary edges connecting with the extra node, the other two edges connect the extra node with all the head nodes of the original two edges, respectively.
- Equivalent: create a relay node on the two edges, respectively, and then merge the two relay nodes.

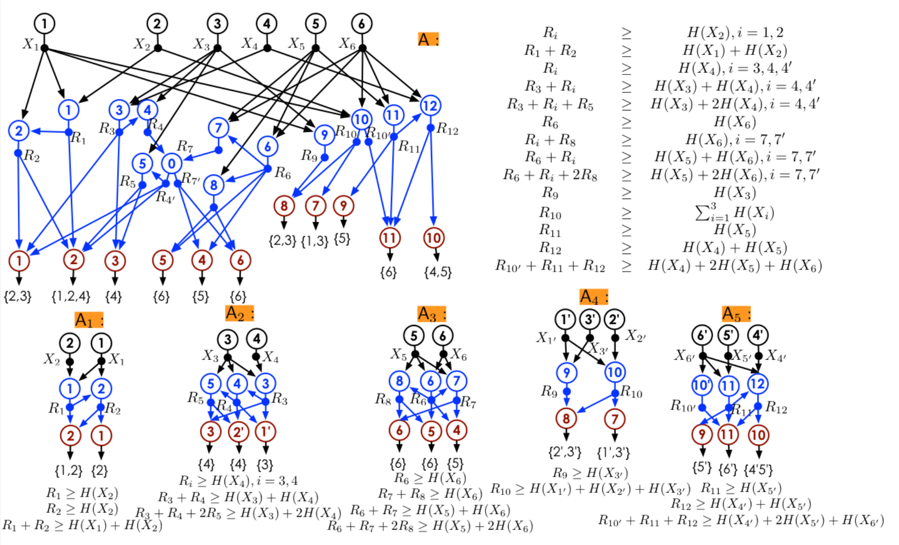


Source merge

- A is obtained by merging $\mathsf{A}_1.\hat{\mathcal{S}} = \mathsf{A}_2.\pi(\hat{\mathcal{S}})$, then for each $i \in \{*, q, (s,q), o\}$
- $\mathcal{R}_i(\mathsf{A}) = \mathrm{Proj}_{\setminus \pi(\hat{\mathcal{S}})}((\mathcal{R}_i(\mathsf{A}_1) \times \mathcal{R}_i(\mathsf{A}_2)) \cap \mathcal{L}_0)$,
- $\mathcal{L}_0 = \left\{ H(Y_s) = H(Y_{\pi(s)}), \forall s \in \hat{\mathcal{S}} \right\}$
- Remark: essentially replace variables $Y_{\pi(s)}$ with the Y_s for each $s \in \hat{\mathcal{S}}$.



Obtain Rate Region for Larger Networks

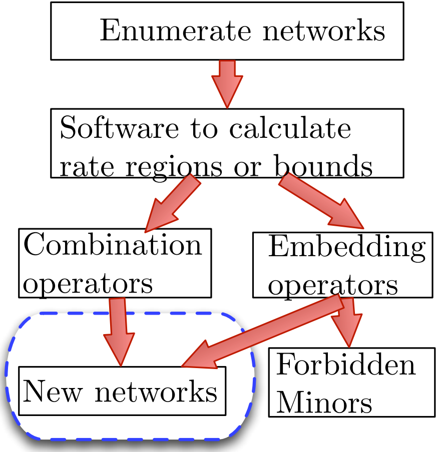


Summary for combination operators

- Four combination operators: source, sink, node, and edge merge
- Rate region of combined network derivable from regions of smaller networks in the combination

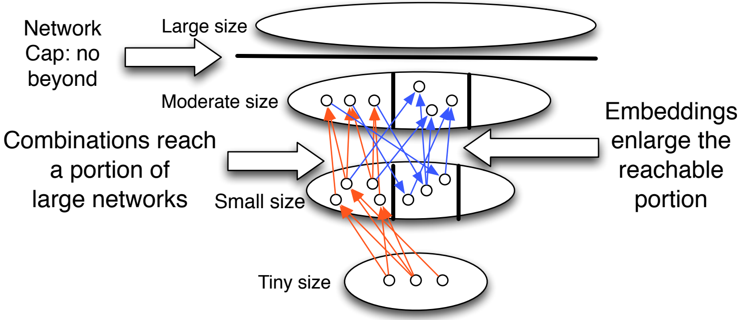
Outline

1. Enumeration
   - Network Model
   - Minimality
   - Network Equivalence Class
   - Enumeration algorithm
2. Rate Region Computation
3. Network Operations
   - Embedding Operations
   - Combination Operations
   - Play with both



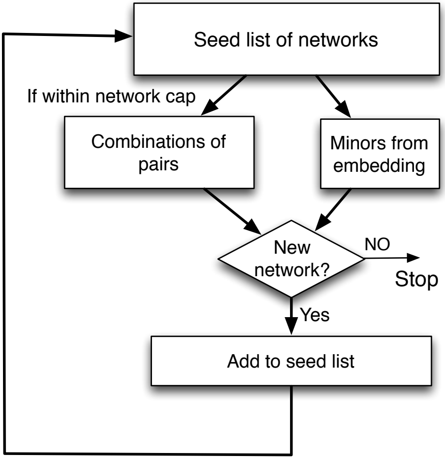
Combination operations suffice?

- Answer is NO.
- Need cap to limit the network size in the combinations.



Partial closure of networks

- Worst case partial closure of networks: cap the predicted size of networks involved in the process
- Let the pool produce new networks until no new network can be generated



New networks found from a tiny seed list

Start with the single $(1,1)$, single $(2,1)$, and the four $(1,2)$ networks;

size\cap	combination operators only			embedding and combinations		
	(3,3)	(3,4)	(4,4)	(3,3)	(3,4)	(4,4)
(1,3)	4	4	4	4	4	4
(1,4)	0	10	10	0	10	10
(2,2)	3	3	3	8	15	16
(2,3)	13	16	16	30	131	155
(2,4)	0	97	101	0	516	648
(3,2)	2	3	2	4	10	11
(3,3)	24	24	24	42	353	833
(3,4)	0	135	135	0	2361	5481
(4,2)	0	0	3	0	0	3
(4,3)	0	0	17	0	0	44
(4,4)	0	0	253	0	0	4430
all	46	292	568	88	3400	11635

New networks found from a tiny seed list

6 tiny networks can generate new 11635 networks with small cap!

size\cap	combination operators only			embedding and combinations		
	(3,3)	(3,4)	(4,4)	(3,3)	(3,4)	(4,4)
(1,3)	4	4	4	4	4	4
(1,4)	0	10	10	0	10	10
(2,2)	3	3	3	8	15	16
(2,3)	13	16	16	30	131	155
(2,4)	0	97	101	0	516	648
(3,2)	2	3	2	4	10	11
(3,3)	24	24	24	42	353	833
(3,4)	0	135	135	0	2361	5481
(4,2)	0	0	3	0	0	3
(4,3)	0	0	17	0	0	44
(4,4)	0	0	253	0	0	4430
all	46	292	568	88	3400	$\boxed{11635}$

New networks found from a tiny seed list

With the increase of cap size, number of new networks increases!

size\cap	combination operators only			embedding and combinations		
	(3,3)	(3,4)	(4,4)	(3,3)	(3,4)	(4,4)
(1,3)	4	4	4	4	4	4
(1,4)	0	10	10	0	10	10
(2,2)	3	3	3	8	15	16
(2,3)	13	16	16	30	131	155
(2,4)	0	97	101	0	516	648
(3,2)	2	3	2	4	10	11
(3,3)	24	24	24	42	353	833
(3,4)	0	135	135	0	2361	5481
(4,2)	0	0	3	0	0	3
(4,3)	0	0	17	0	0	44
(4,4)	0	0	253	0	0	4430
all	46	292	568	88	3400	11635

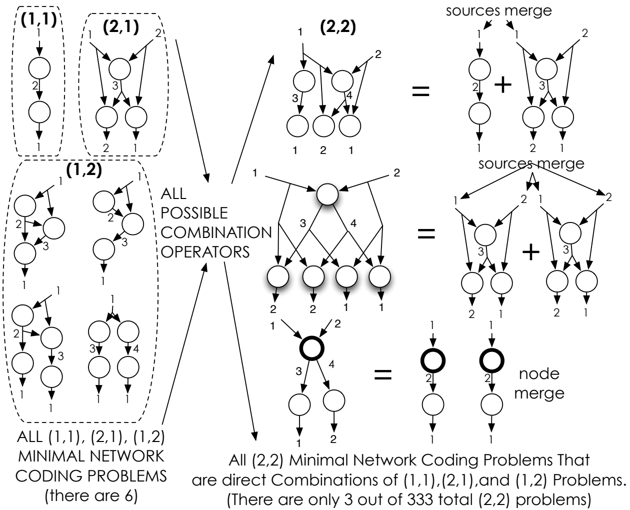
New networks found from a tiny seed list

Embedding operations are important in the process!

size\cap	combination operators only			embedding and combinations		
	(3,3)	(3,4)	(4,4)	(3,3)	(3,4)	(4,4)
(1,3)	4	4	4	4	4	4
(1,4)	0	10	10	0	10	10
(2,2)	3	3	3	8	15	16
(2,3)	13	16	16	30	131	155
(2,4)	0	97	101	0	516	648
(3,2)	2	3	2	4	10	11
(3,3)	24	24	24	42	353	833
(3,4)	0	135	135	0	2361	5481
(4,2)	0	0	3	0	0	3
(4,3)	0	0	17	0	0	44
(4,4)	0	0	253	0	0	4430
all	46	292	568	88	3400	11635

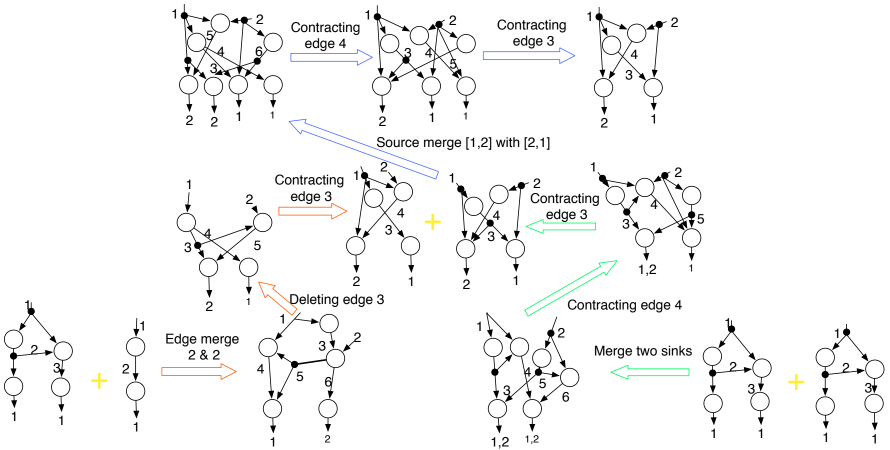
Example to see why integrating embedding is important

- Start with the single $(1,1)$, single $(2,1)$, and the four $(1,2)$ networks
- Only combination with cap $(3,4)$, get only 3 networks with size $(2,2)$



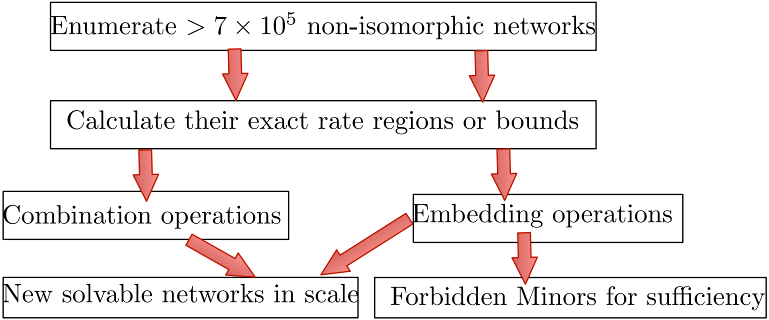
Example to see why integrating embedding is important

- Start with the single $(1,1)$, single $(2,1)$, and the four $(1,2)$ networks
- Consider both combination and embedding with same cap, found $(2,2)$ networks unreachable by combination only



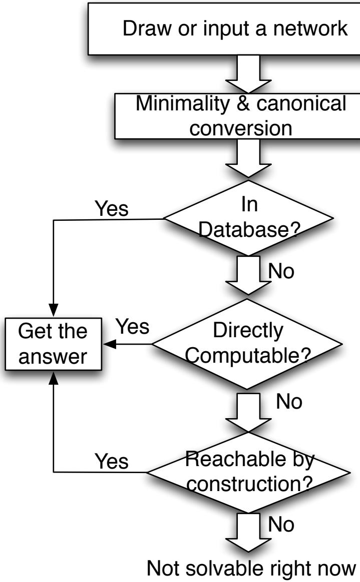
Summary of work thus far

Enumeration, Rate Region Computation, Forbidden Minors, New Networks



Dream

- Though we have online repository
- Want a user-friendly interface to easily get answer



Future work

- When Shannon outer bound is tight? Any common structure?
  - Is Shannon outer bound tight for all MDCS, or IDSC?
- Is the number of forbidden minors regarding the sufficiency of a class of linear codes finite?
- Coverage of the operators in all problems
- More operations: node & edge merge, source & sink merge
- A notion of forbidden minor which can harness both combination and embedding operators

Selected References

X. Yan, R.W. Yeung, and Z. Zhang (2012)
An Implicit Characterization of the Achievable Rate Region for Acyclic Multisource Multisink Network Coding
IEEE Transactions on Information Theory 58(9), 5625-5639.

A. Betten, M. Braun, H. Fripertinger (2006)
Error-Correcting Linear Codes: Classification by Isometry and Applications
Springer Berlin Heidelberg.

N. Robertson, P. Seymour (2004)
Graph minors. xx. wagner's conjecture
Journal of Combinatorial Theory, Series B, 92(2), 325-357.

J.G. Oxley (2011)
Matroid Theory
Oxford University, 2011.

Q & A

Thank you!