Network Coding: Computation, Symmetry, and Hierarchy

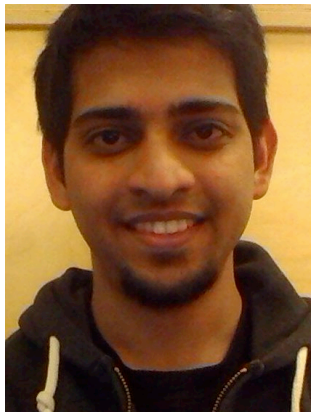
John MacLaren Walsh
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, PA
jwalsh@ece.drexel.edu

Thanks to NSF CCF-1421828 & NSF CCF-1053702.

Collaborators & Co-Authors

Congduan Li, Ph.D.
Postdoctoral Fellow,
Institute for Network Coding
Chinese Univ. of Hong Kong

rate region database
network operators
forbidden net. minors



Jayant Apte,
Ph.D. Candidate,
ASPITRG,
Drexel University

network symmetry
software: ITAP & CHM

Steven Weber, Ph.D.
Professor, Dept. of ECE
Drexel University

co-advisor for
Congduan Li

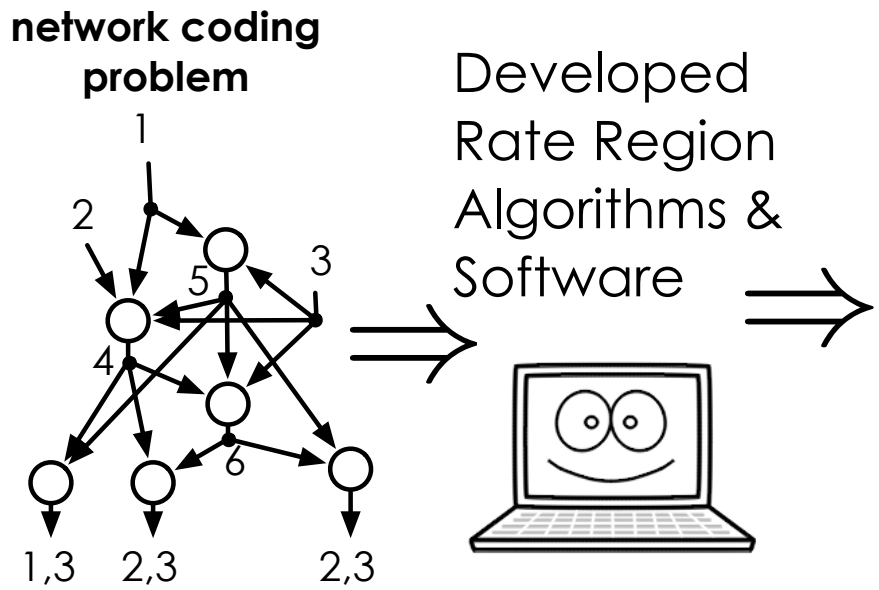
Yunshu Liu,
Ph.D. Candidate
ASPITRG,
Drexel University

nonlinear entropic
vectors & codes

Outline

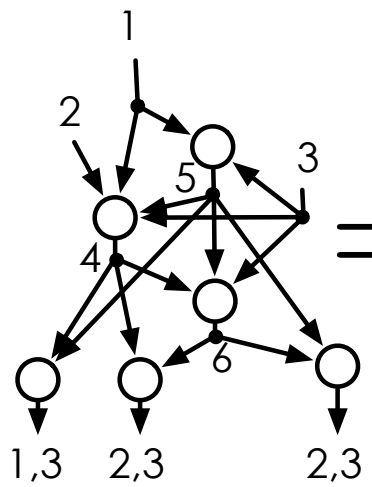
1. Problem Statement. Minimality & Symmetry
2. Symmetry Exploiting Rate Region Computation
   (a) Converse: Symmetry Exploiting CHM
   (b) Achievability:
      i. Linear Code Enumeration
      ii. Nonlinear Code (Partition) Enumeration
3. Rate Region Database
4. Hierarchy: Network Operators
   (a) Big $\rightarrow$ Small: Embedding Operators
   (b) Small $\rightarrow$ Big: Combinations Operators
   (c) Operator Concatenation
   (d) Coverage

Computationally Enabled Research Agenda – Motivation



Computationally Enabled Research Agenda – Motivation

network coding problem



Developed Rate Region Algorithms & Software

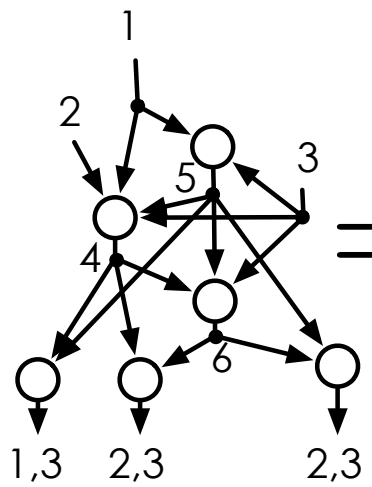
inequality description

$$\min\{R_4, R_6\} \geq H(Y_2)$$
$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$
$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$
$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$
$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

rate region

Computationally Enabled Research Agenda – Motivation

network coding problem



Developed Rate Region Algorithms & Software

inequality description

$$\min\{R_4, R_6\} \geq H(Y_2)$$
$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$
$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$
$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$
$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

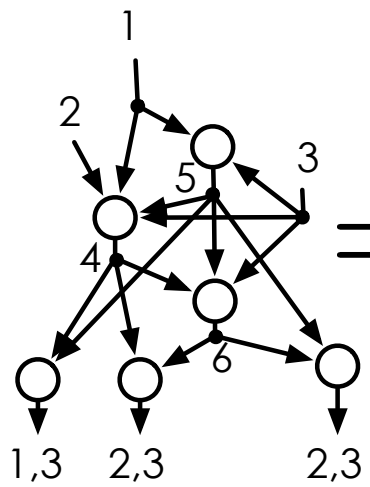
extreme ray description

$H(Y_1)$	1	0	0	0	0	0	0	1	0	0	0
$H(Y_2)$	0	1	0	0	0	0	1	0	0	0	0
$H(Y_3)$	0	1	2	1	1	0	0	0	0	1	0
R_4	0	1	1	1	1	1	1	1	0	0	0
R_5	1	1	1	0	1	0	0	0	0	1	1
R_6	0	1	1	1	0	0	1	0	1	1	0

rate region

Computationally Enabled Research Agenda – Motivation

network coding problem



Developed Rate Region Algorithms & Software

inequality description

$$\min\{R_4, R_6\} \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$

$$\boxed{R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)}$$

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

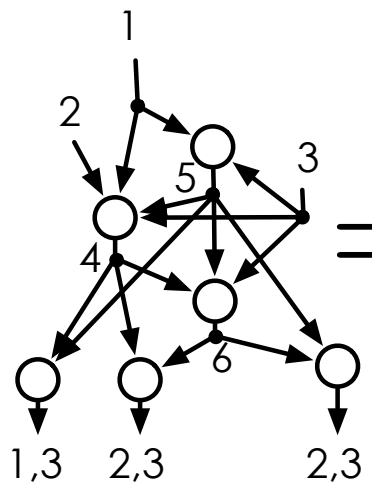
extreme ray description

$H(Y_1)$	1	0	0	0	0	0	0	1	0	0	0
$H(Y_2)$	0	1	0	0	0	0	1	0	0	0	0
$H(Y_3)$	0	1	2	1	1	0	0	0	0	1	0
R_4	0	1	1	1	1	1	1	1	0	0	0
R_5	1	1	1	0	1	0	0	0	0	1	1
R_6	0	1	1	1	0	0	1	0	1	1	0

rate region

Computationally Enabled Research Agenda – Motivation

network coding problem



Developed Rate Region Algorithms & Software

automated converse proofs

inequality description

$$\min\{R_4, R_6\} \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

Proof:

$$R_4 \geq H(U_4)$$

●
●
●

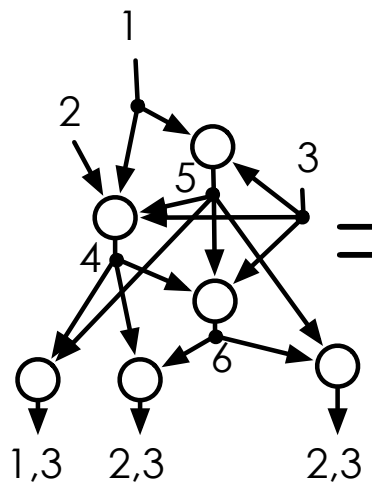
extreme ray description

$H(Y_1)$	1	0	0	0	0	0	0	1	0	0	0
$H(Y_2)$	0	1	0	0	0	0	1	0	0	0	0
$H(Y_3)$	0	1	2	1	1	0	0	0	0	1	0
R_4	0	1	1	1	1	1	1	1	0	0	0
R_5	1	1	1	0	1	0	0	0	0	1	1
R_6	0	1	1	1	0	0	1	0	1	1	0

rate region

Computationally Enabled Research Agenda – Motivation

network coding problem



Developed Rate Region Algorithms & Software

automated converse proofs

inequality description

$$\min\{R_4, R_6\} \geq H(Y_2)$$

$$R_4 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_5 + R_6 \geq H(Y_2) + H(Y_3)$$

$$R_4 + R_5 \geq H(Y_1) + H(Y_2) + H(Y_3)$$

$$R_4 + R_5 + 2R_6 \geq H(Y_1) + 2H(Y_2) + 2H(Y_3)$$

Proof:

$$R_4 \geq H(U_4)$$

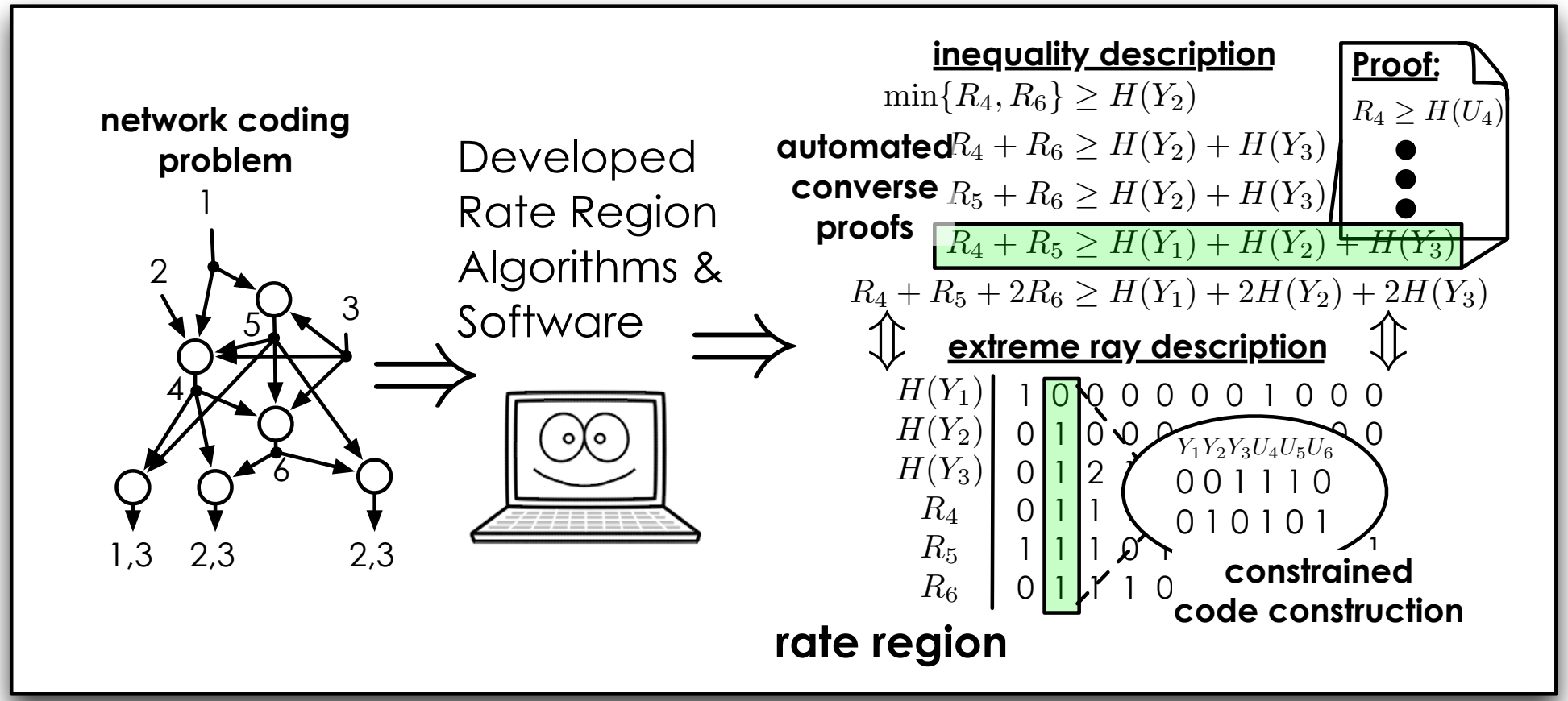
●
●
●

extreme ray description

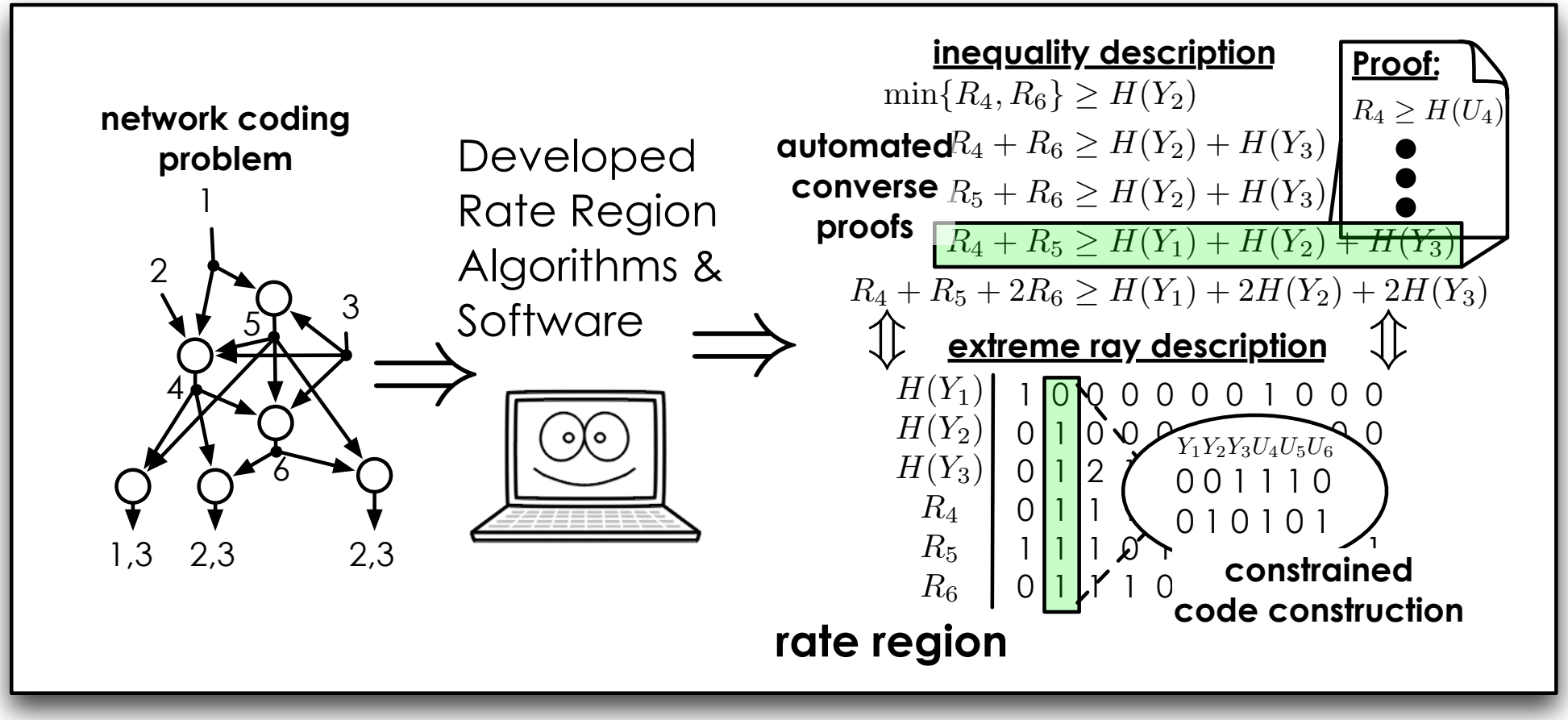
$H(Y_1)$	1	0	0	0	0	0	1	0	0	0
$H(Y_2)$	0	1	0	0	0	1	0	0	0	0
$H(Y_3)$	0	1	2	1	1	0	0	0	1	0
R_4	0	1	1	1	1	1	1	0	0	0
R_5	1	1	1	0	1	0	0	0	1	1
R_6	0	1	1	1	0	0	1	0	1	1

rate region

Computationally Enabled Research Agenda – Motivation



Computationally Enabled Research Agenda – Motivation



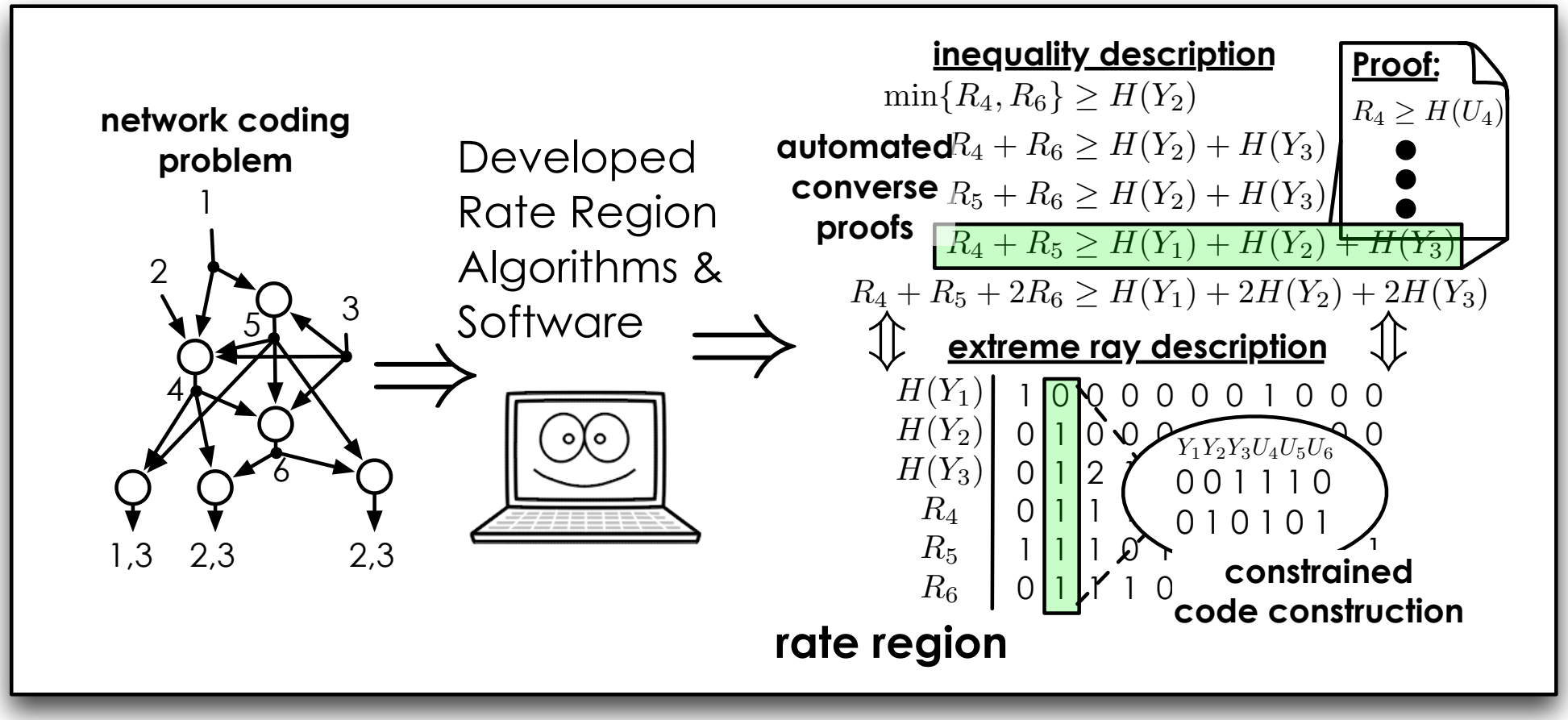
Substitutes outer/inner bounds to Γ_N^* into Yan, Yeung, Zhang '12

$$\mathcal{R}_* = \text{proj}_{R_\mathcal{E}, [H(Y_s)|s\in\mathcal{S}]}(\overline{\text{con}(\Gamma_\text{N}^* \cap \mathcal{L}_{123})} \cap \mathcal{L}_{45}) \tag{1}$$

where $\mathcal{L}_{123} := \left\{ \mathbf{h} \,\middle|\, h_{Y_\mathcal{S}} = \Sigma_{s\in\mathcal{S}} h_{Y_s},\ h_{X_{\text{Out}(k)}|Y_s} = 0,\ h_{X_{\text{Out}(i)}|X_{\text{In}(i)}} = 0 \right\}$ and

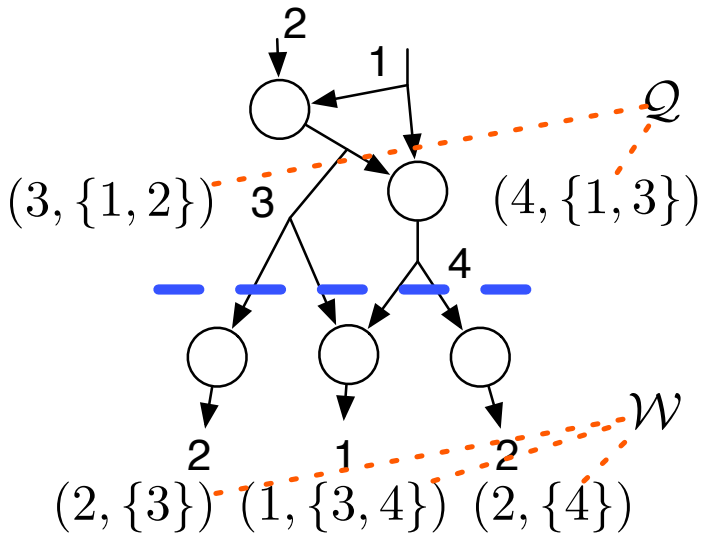
$\mathcal{L}_{45} := \{(\mathbf{h}^T, \mathbf{R}^T)^T \in \mathbb{R}_+^{2^N-1+|\mathcal{E}|} : R_e \geq h_{U_e}, e \in \mathcal{E},\ h_{Y_{\beta(t)}|U_{\text{In}(t)}} = 0\}$

Computationally Enabled Research Agenda – Motivation



- Want software to reach as large networks as possible. $\Longrightarrow$ Define & Exploit Problem Symmetry

Exploiting Symmetry in Rate Region Computation – Definition [Li Arxiv '15]



- Represent a network as $\mathsf{A} := (\mathcal{Q}, \mathcal{W})$:
  - sources: $1 \dots K$
  - edges: $K+1, \dots, K+|\mathcal{E}_U|$
  - Edge definitions $\mathcal{W}$ a set of $(i, \mathcal{A})$, $i \in \{K+1, \dots, K+|\mathcal{E}_U|\}$, $\mathcal{A} \subseteq \{1, \dots, K+|\mathcal{E}_U|\} \setminus \{i\}$;
  - Sink definitions $\mathcal{Q}$ a set of $(i, \mathcal{A})$, $i \in \{1, \dots, K\}$, $\mathcal{A} \subseteq \{1, \dots, K+|\mathcal{E}_U|\} \setminus \{i\}$;
  - Same i can appear in $\mathcal{W}$ but not in $\mathcal{Q}$
- Group $\mathcal{G} := S_{\{1,2,\dots,K\}} \times S_{\{K+1,\dots,K+|\mathcal{E}_U|\}}$ acts naturally on $(\mathcal{Q}, \mathcal{W})$.
- Network Symmetry Group (NSG): stabilizer subgroup $\mathcal{G}_\mathsf{A} = \{g \in \mathcal{G} | g((\mathcal{Q}, \mathcal{W})) = (\mathcal{Q}, \mathcal{W})\}$.
- NSG Stabilizes constraints $\mathcal{L}_{123}$ and $\mathcal{L}_{45}$ in rate region expression. $\pi(\mathcal{R}_*(\mathsf{A})) = \mathcal{R}_*(\mathsf{A})$ for all $\pi \in \mathcal{G}_\mathsf{A}$
- any $g \in \mathcal{G}$ w/ $g \notin \mathcal{G}_\mathsf{A}$ gives an equivalent network $\mathsf{A}' = g\mathsf{A}$ s.t. $g(\mathcal{R}(\mathsf{A})) = \mathcal{R}_*(\mathsf{A}')$.
- Orbit Stabilizer Theorem $\implies$ there are $\frac{|\mathcal{G}|}{|\mathcal{G}_\mathsf{A}|}$ such equivalent networks

Exploiting Symmetry in Rate Region Computation – Converse

Converse: calculate

$$\mathcal{R}_o = \text{proj}_{R_\mathcal{E},[H(Y_s)|s\in\mathcal{S}]}(\Gamma_N^o \cap \mathcal{L}_{12345}) \tag{2}$$

through polyhedral projection.

Extreme Ray/Point Projection

Project Extreme Rays & Points

↓

Redundancy Removal (convex hull)

extremal repr. of projected polyhedron

↓

Representation conversion

↓

inequality repr. of projected polyhedron

Fourier Motzkin Elimination

Successively Eliminates Variables (slow)

inequalities of N dimen. polyhedron

↓

N-1 dim. proj. (elim. 1 var)

↓

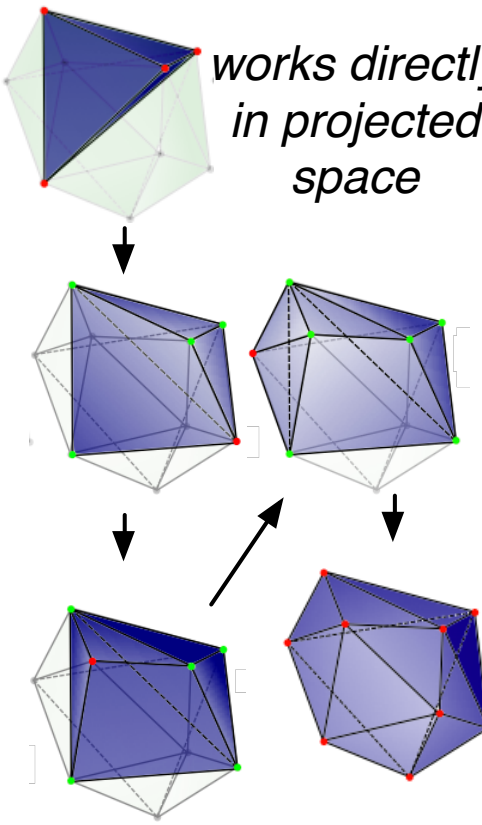
N-2 dim. proj. (elim. 1 var)

↓

⋮

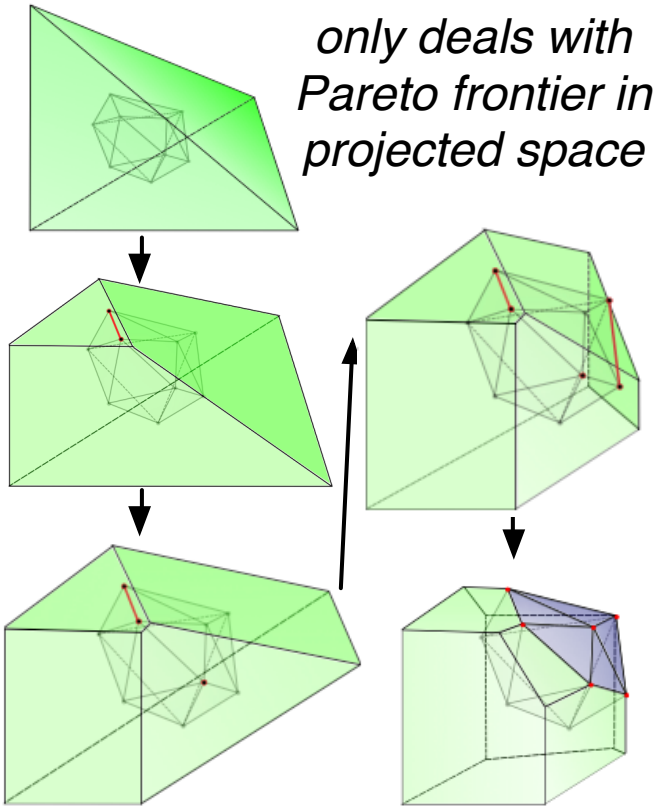
Convex Hull Method

works directly in projected space

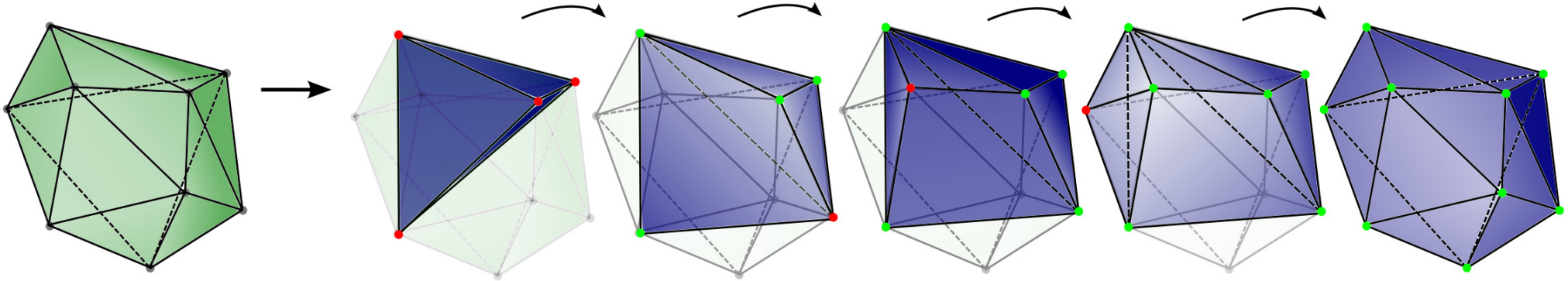


Benson's Outer Approximation Algorithm

only deals with Pareto frontier in projected space

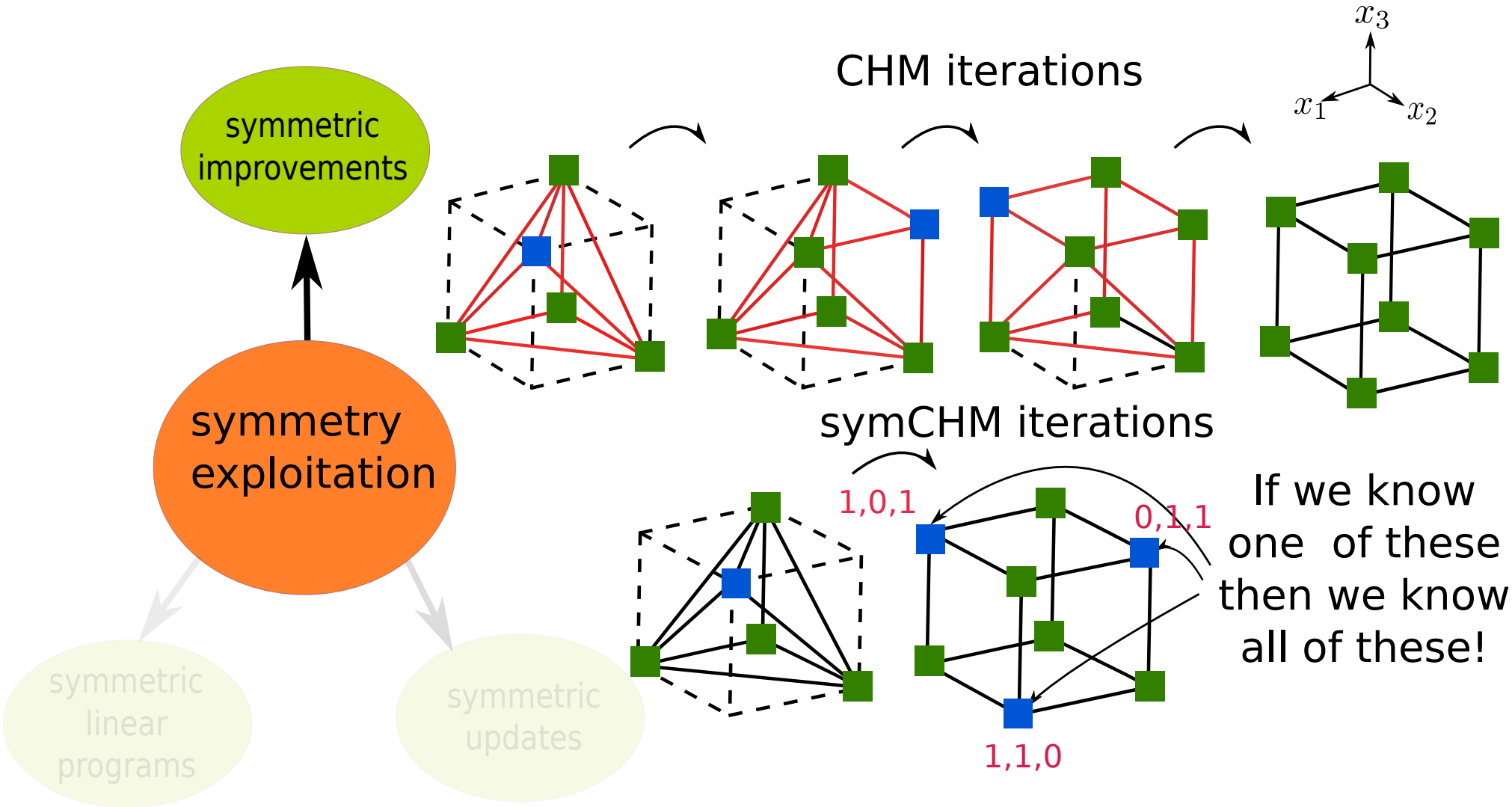


Exploiting Symmetry in Rate Region Computation – Converse



Apte's CHM w/ rational LP solver implementation available from ASPITRG website.

Exploiting Symmetry in Rate Region Computation – Converse [Apte Netcod 2015]



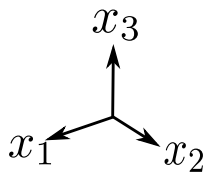
Exploiting Symmetry in Rate Region Computation – Converse [Apte Netcod 2015]

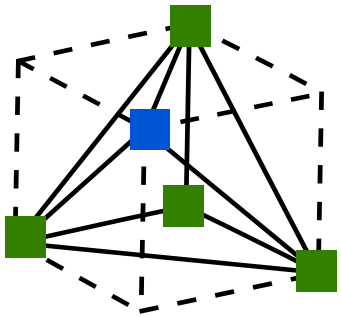
symmetric improvements

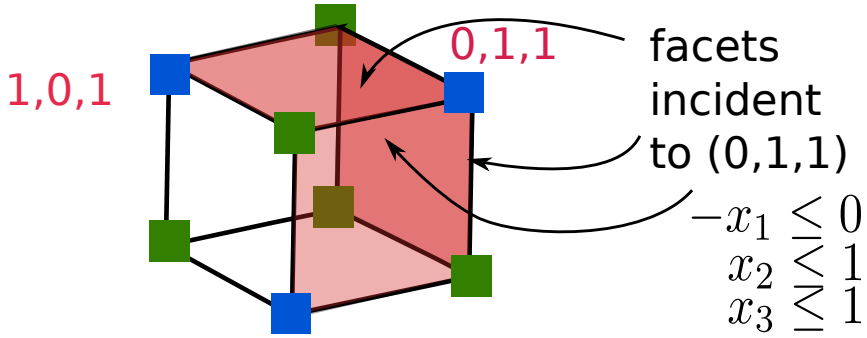
symmetry exploitation

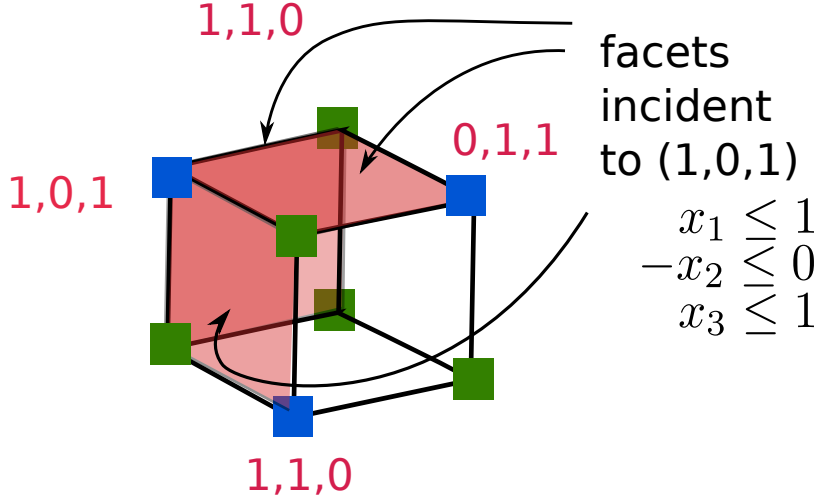
symmetric linear programs

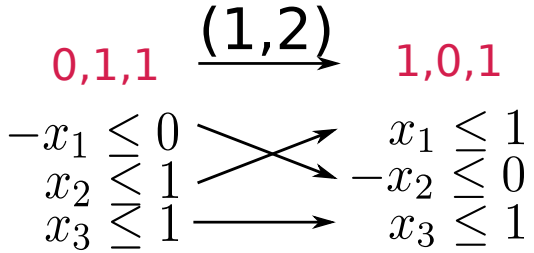
symmetric updates



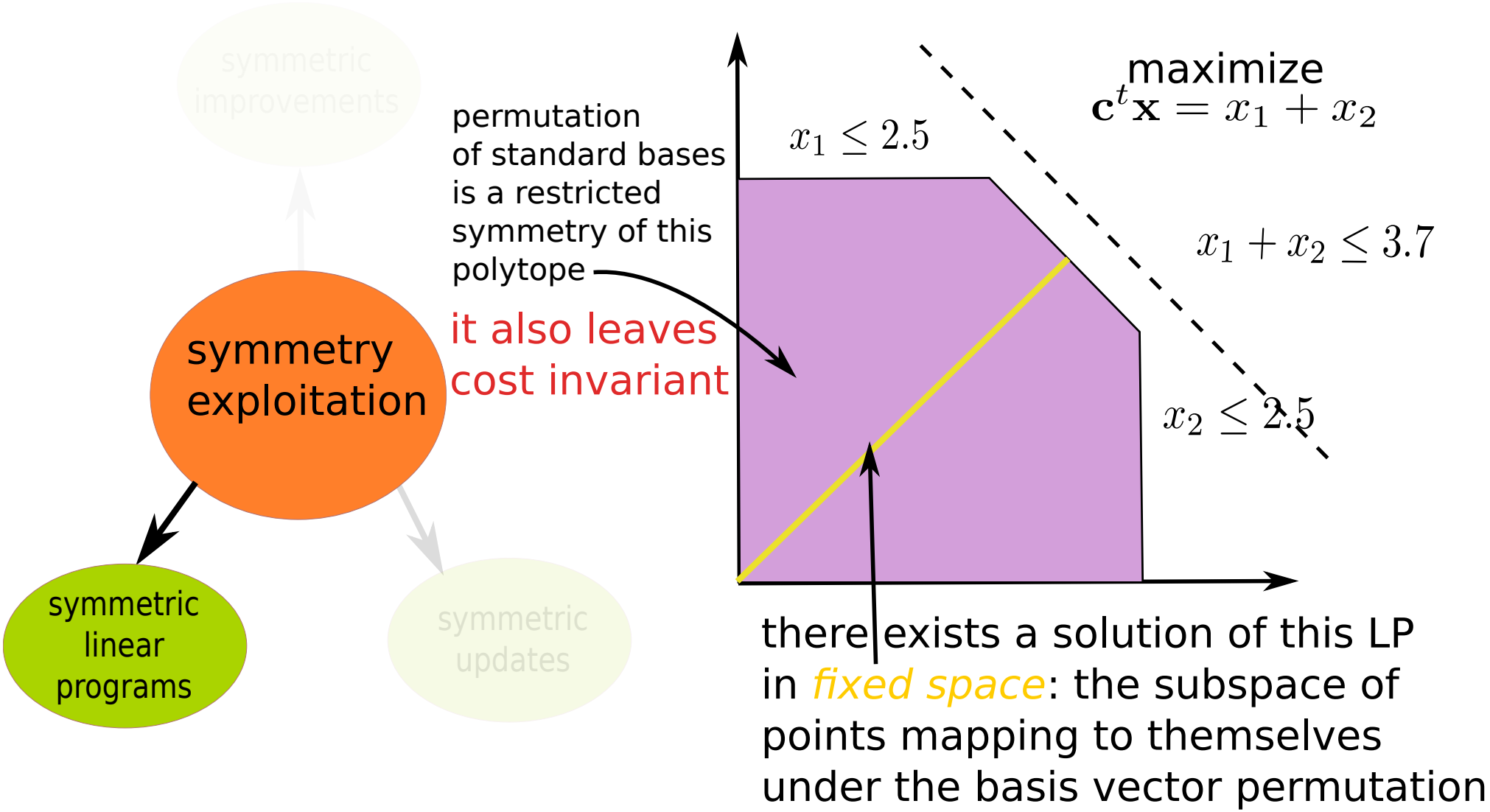




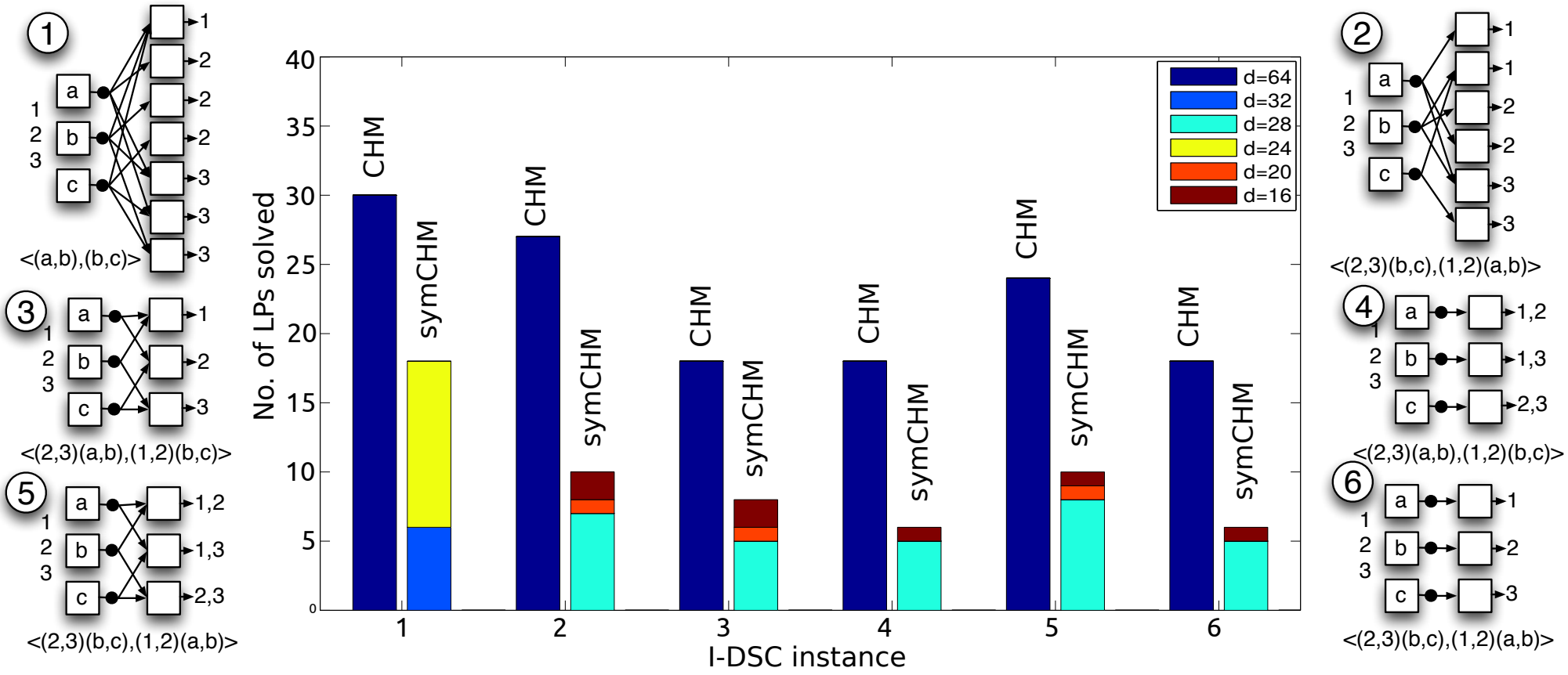




Exploiting Symmetry in Rate Region Computation – Converse [Apte Netcod 2015]



Exploiting Symmetry in Rate Region Computation – Converse [Apte Netcod 2015]



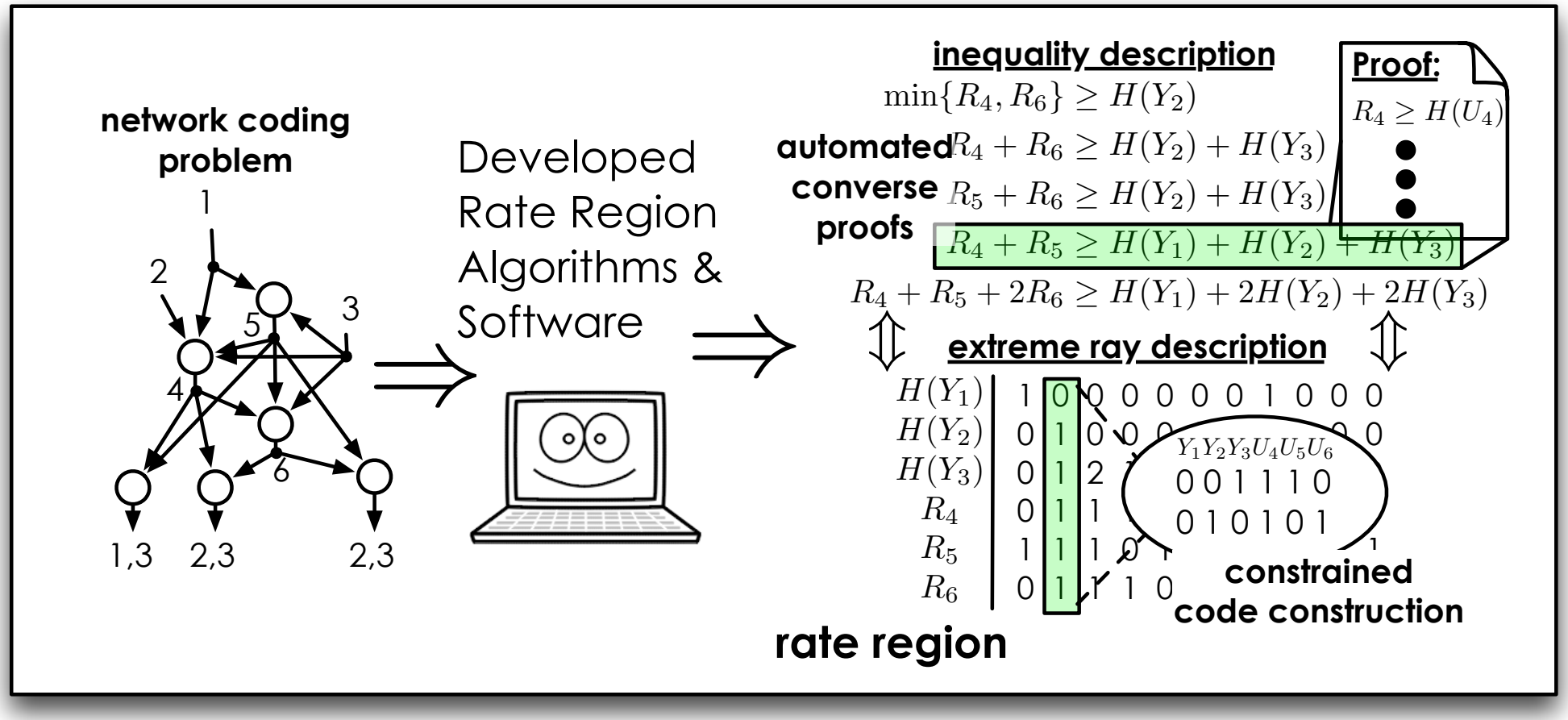
Exploiting Symmetry in Rate Region Computation – Achievability

- Key Idea: linear network code $= (\mathcal{V}, \phi)$
  - $\mathcal{V}$ a *set* of $\ell \leq K + |\mathcal{E}_U|$ subspaces of $GF(q)^r$
  - a map $\phi : \mathcal{V} \to \{1, \ldots, K + |\mathcal{E}_U|\}$ assigning these subspaces to network sources & edges.
- Note: rate region only depends on subset entropies=dimensions/ranks of these subspaces and their sums! (associated polymatroid).
- Two sets $\mathcal{V}$ and $\mathcal{V}'$ will give the same ranks if $\mathcal{V}' = \mathcal{V}g$ for some $g \in P\Gamma L_q^r$!
- Linear Code Enumeration Problem:
  - finding only canonical representatives of orbits in the sets of subspaces $\mathcal{V}$ under $P\Gamma L_q^r$ which can be grouped with a ϕ to obey network constraints (& optionally have specified $R_e, H(Y_s)$ dimension vectors)
  - Algorithm Leiterspiel [Schmalz, Betten Braun & Fripertinger] for finding orbits in the power-set obeying an inherited property recursively in subset-size can be applied to solve this problem.

  - Inherited property: ability to map (partial ϕ or p-map) to a (thus far defined) subset of network variables and obey the constraints they are involved in.
  - NSG (together with stabilizer subgroup of the subspaces $\mathcal{V}$) makes some ϕ equivalent with one another, removing the need to test all of them.

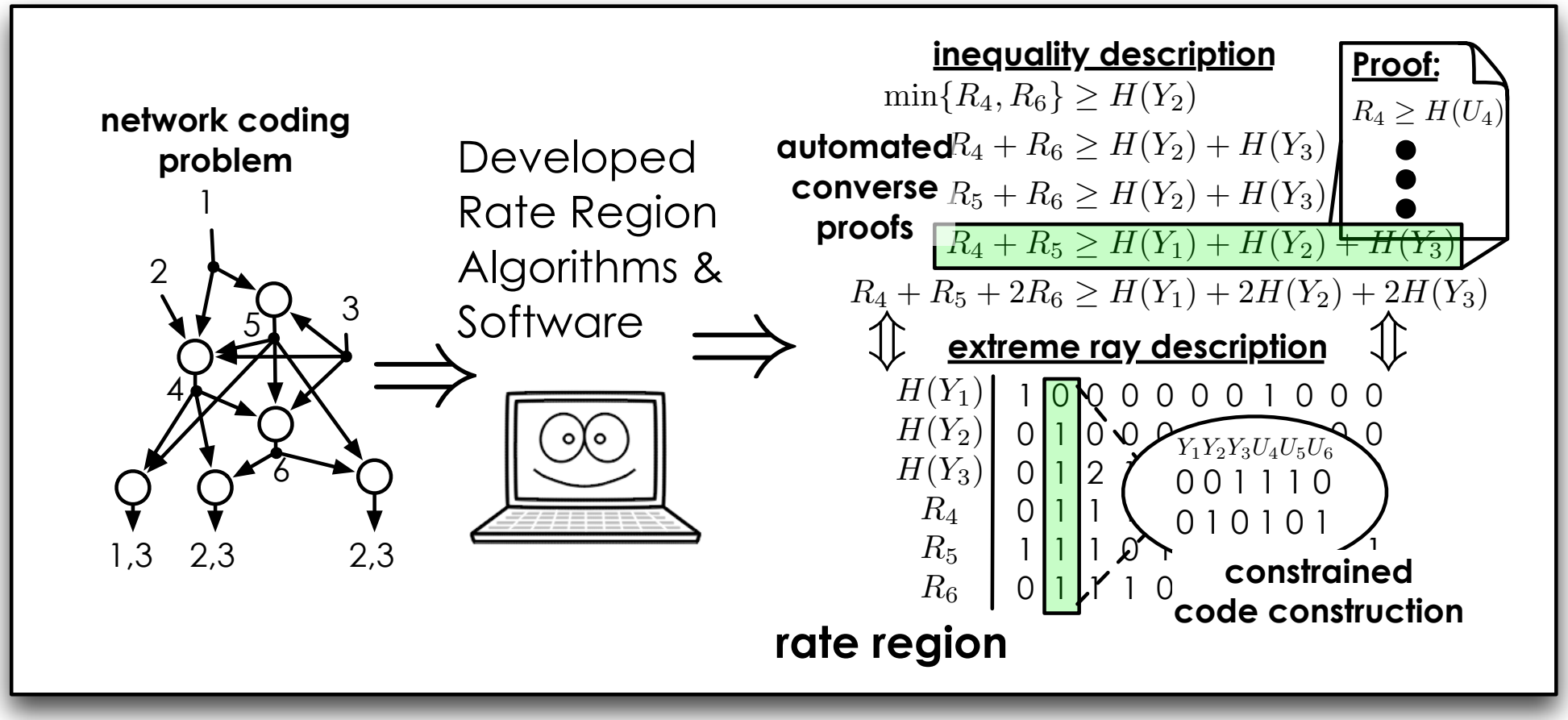
- Method above implemented in our **ITAP** (github) GAP package
- C++ version building on Orbiter [Betten] forthcoming.
- Although enumeration oriented, when used as a verification algorithm (w/ specified rate vector) it can still be faster than the Groebner basis (w/ Singular) based path-gain verification of Subramanian & Thangaraj!
- Nonlinear codes: view discrete random variables (sources & edges) each as partitions of a finite Ω from $(\Omega, \mathcal{F}, \mathbb{P})$. Group is $S_{|\Omega|}$. Have done this for E.V.s [Liu arxiv 2016] but could be coupled with p-maps ϕ to yield nonlinear codes.

Computationally Enabled Research Agenda – Motivation



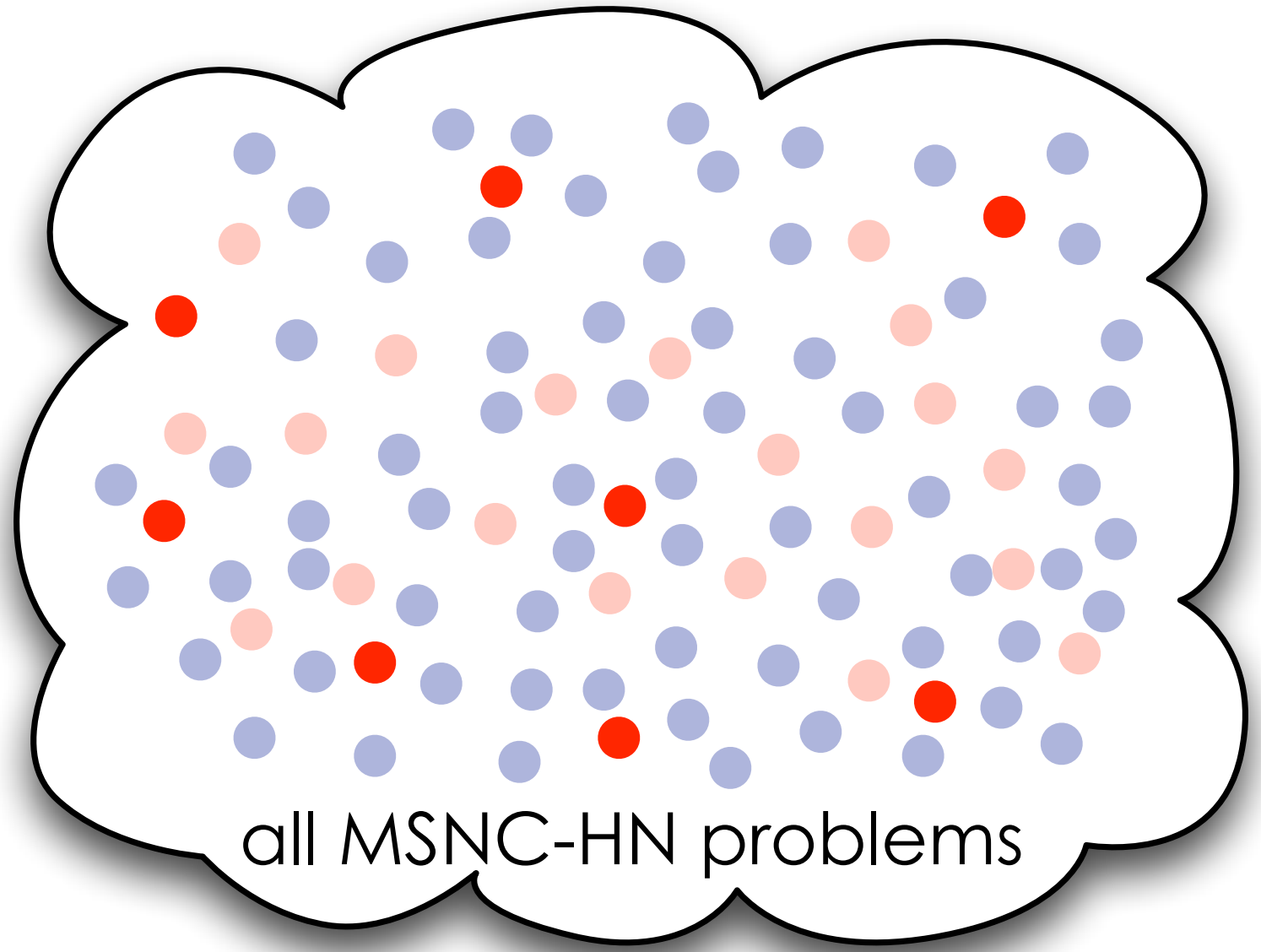
- Want software to reach as large networks as possible. $\Longrightarrow$ Define & Exploit Problem Symmetry

Computationally Enabled Research Agenda – Motivation

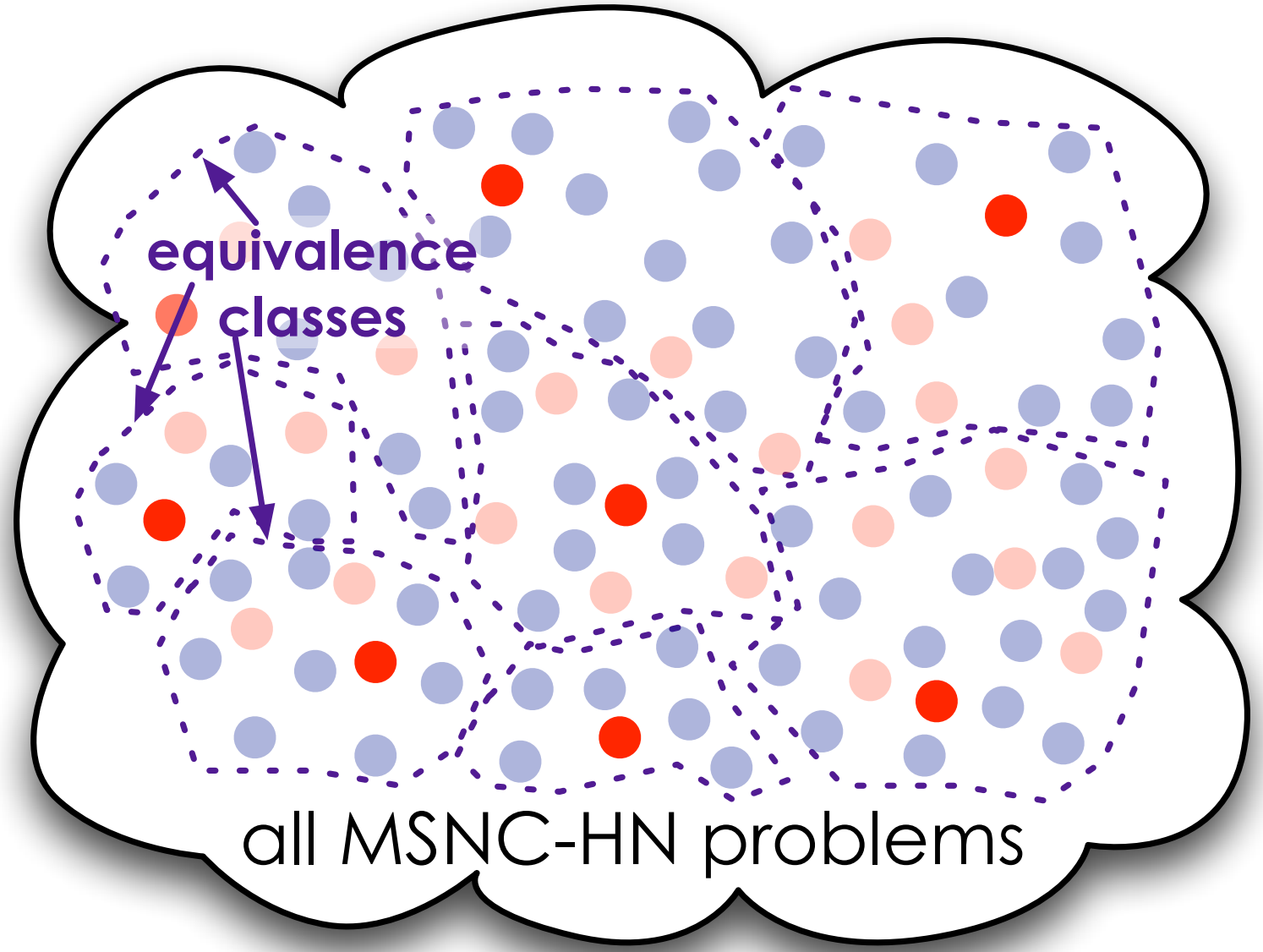


- Want software to reach as large networks as possible. $\Longrightarrow$ Define & Exploit Problem Symmetry
- Want the database of solved problems to be as large as possible. $\Longrightarrow$ Define Problem Equivalence Classes via Canonical & Minimal Representatives

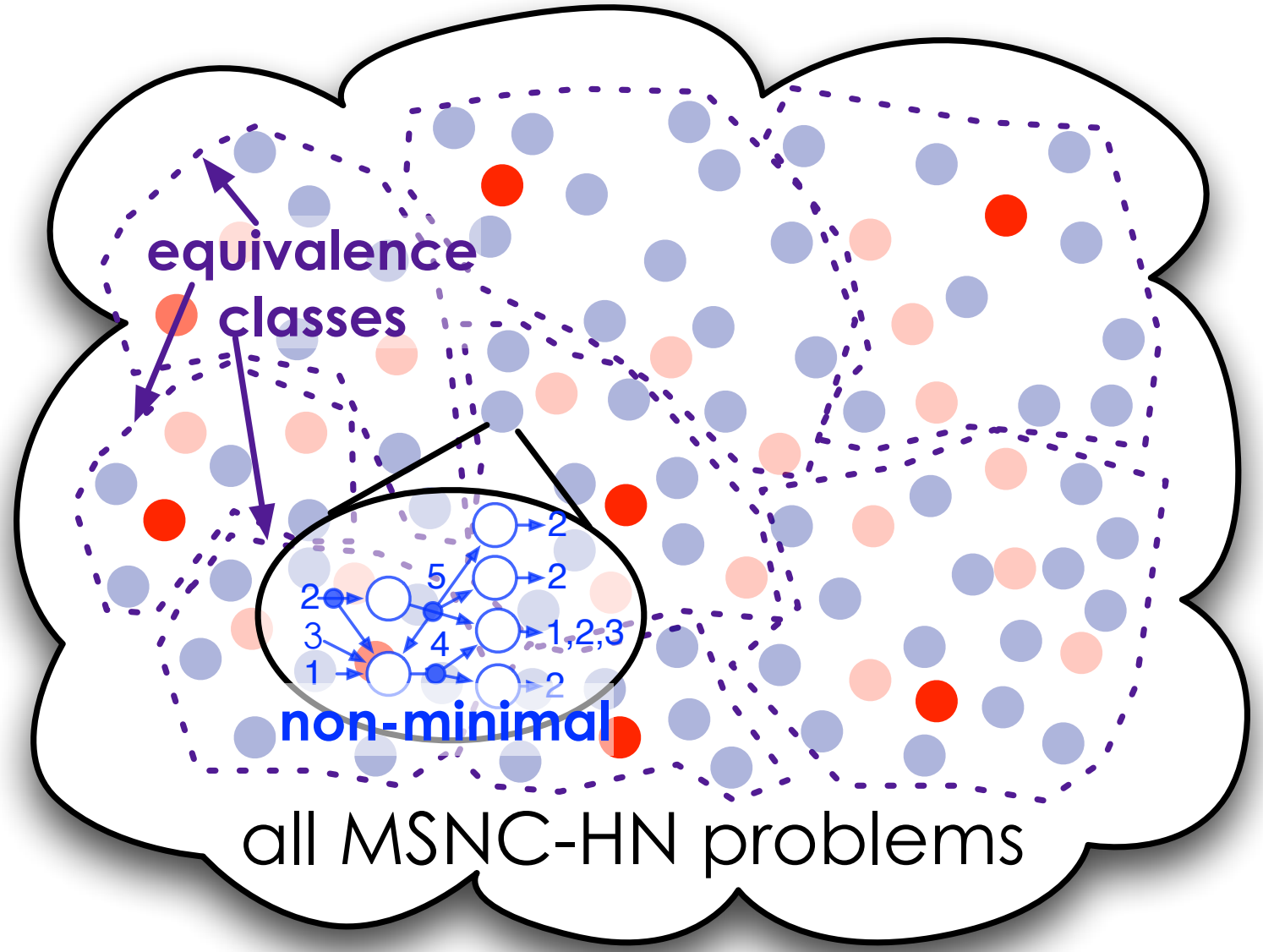
Computationally Enabled Research Agenda – Minimality & Symmetry



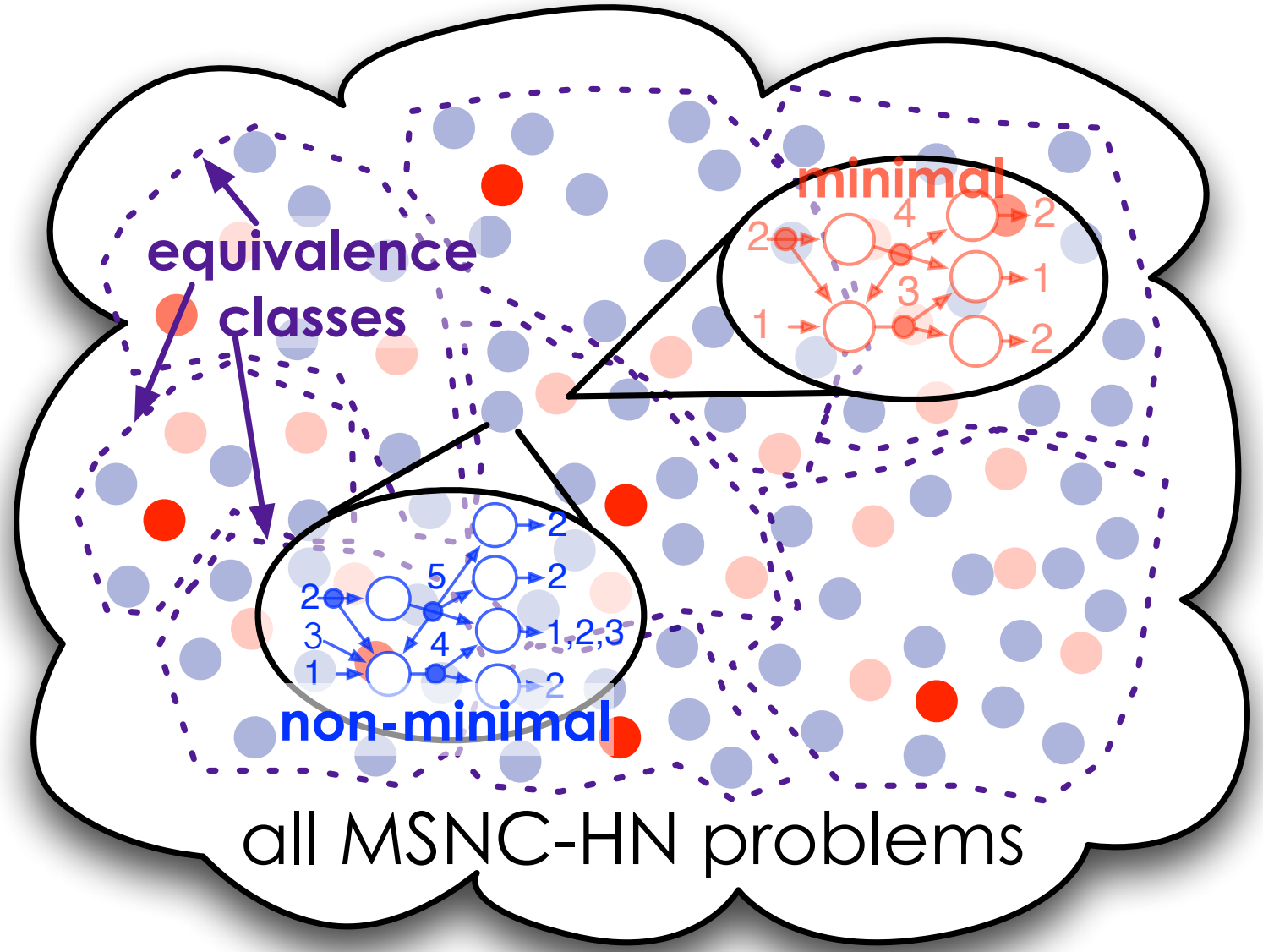
Computationally Enabled Research Agenda – Minimality & Symmetry



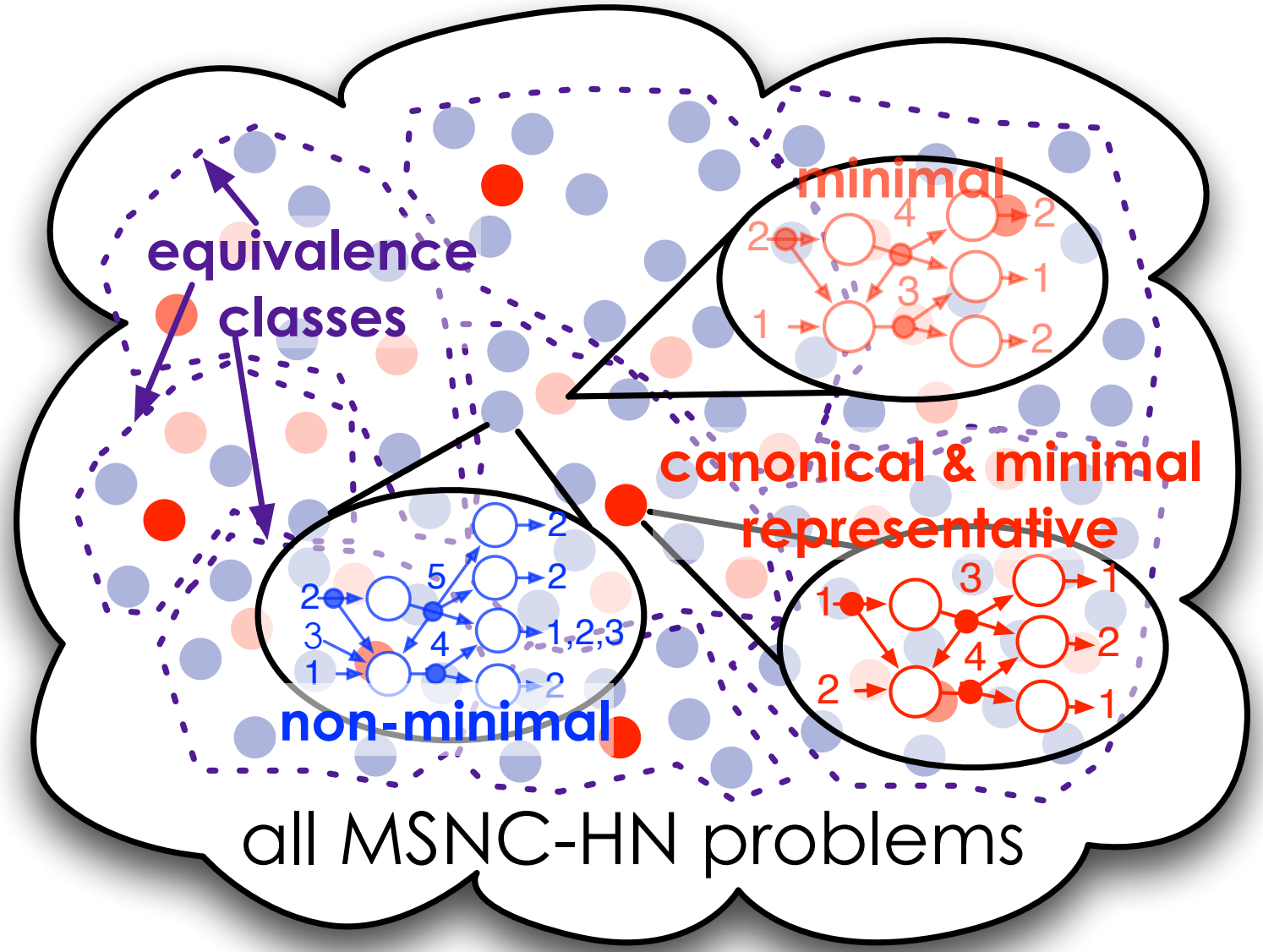
Computationally Enabled Research Agenda – Minimality & Symmetry



Computationally Enabled Research Agenda – Minimality & Symmetry

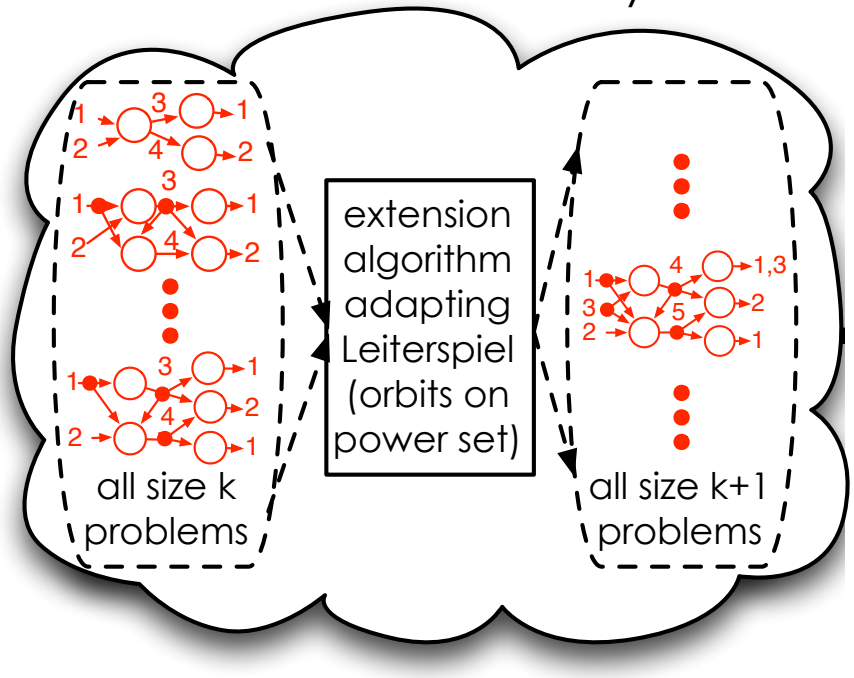


Computationally Enabled Research Agenda – Minimality & Symmetry

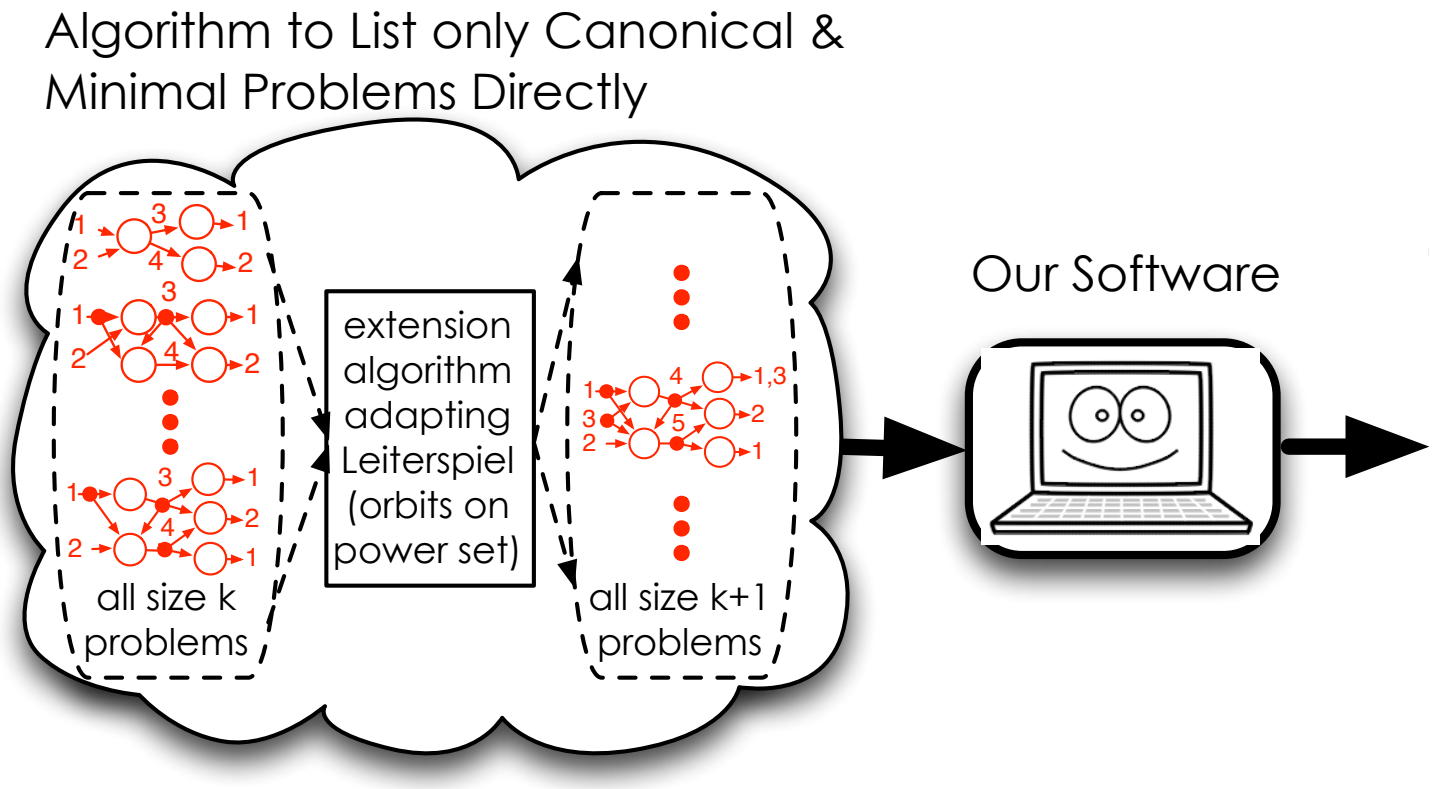


Computationally Enabled Research Agenda – Rate Region Database

Algorithm to List only Canonical & Minimal Problems Directly

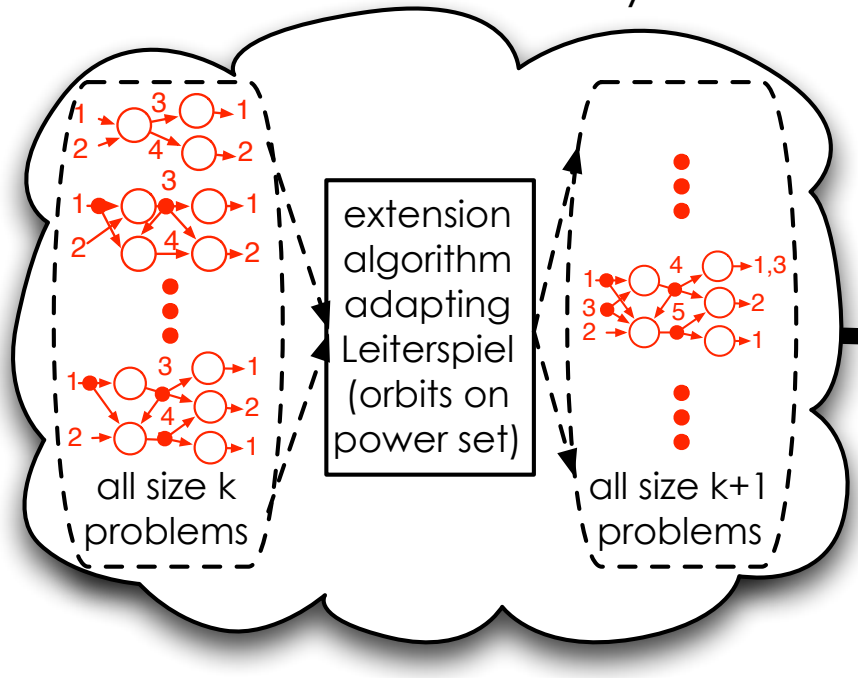


Computationally Enabled Research Agenda – Rate Region Database

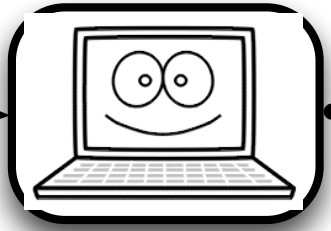


Computationally Enabled Research Agenda – Rate Region Database

Algorithm to List only Canonical & Minimal Problems Directly

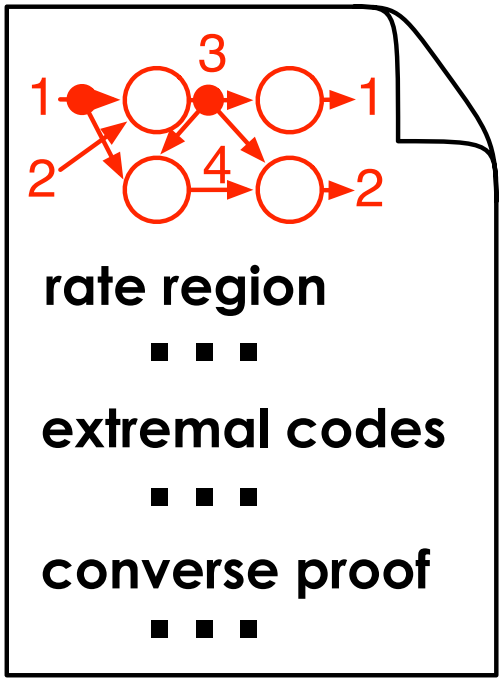


Our Software



Rate Region Database:
rate region,
converse proof,
efficient codes
for every canonical & minimal problem

Network Coding Rate Region Database



rate region

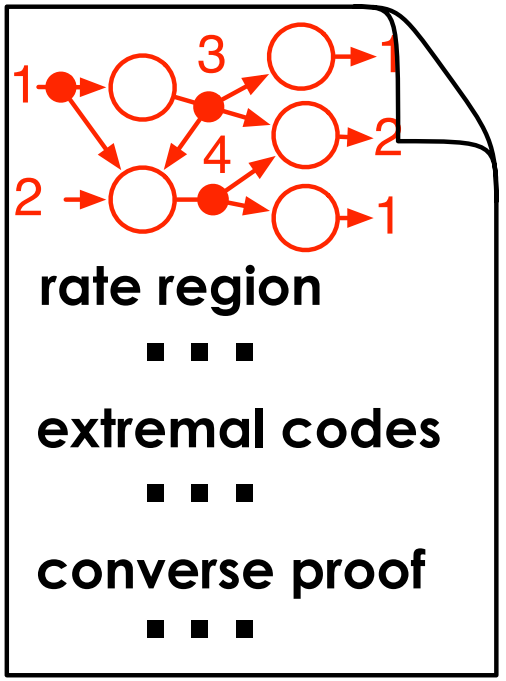
▪ ▪ ▪

extremal codes

▪ ▪ ▪

converse proof

▪ ▪ ▪



rate region

▪ ▪ ▪

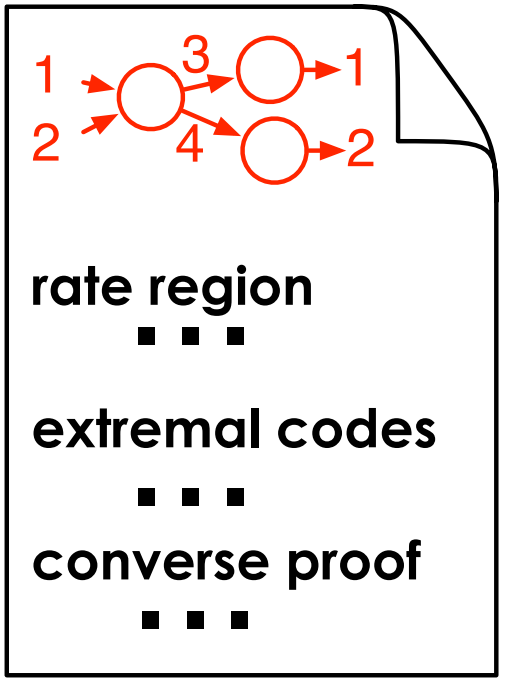
extremal codes

▪ ▪ ▪

converse proof

▪ ▪ ▪

▪ ▪ ▪



rate region

▪ ▪ ▪

extremal codes

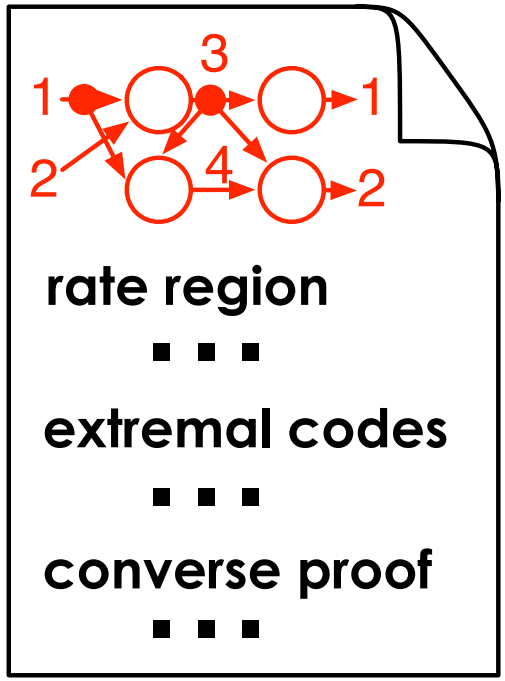
▪ ▪ ▪

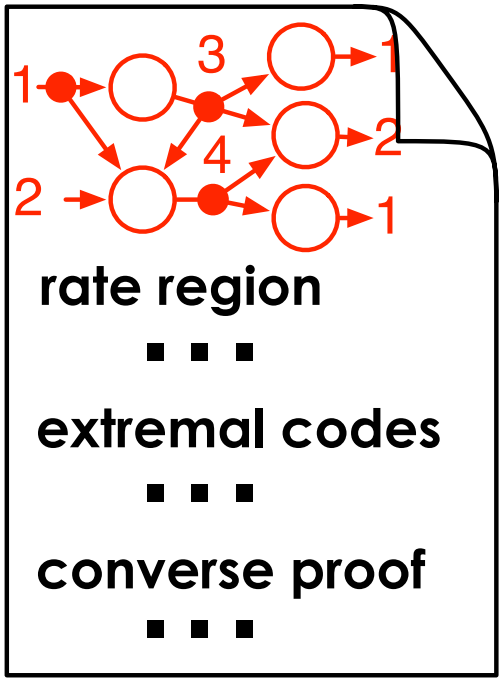
converse proof

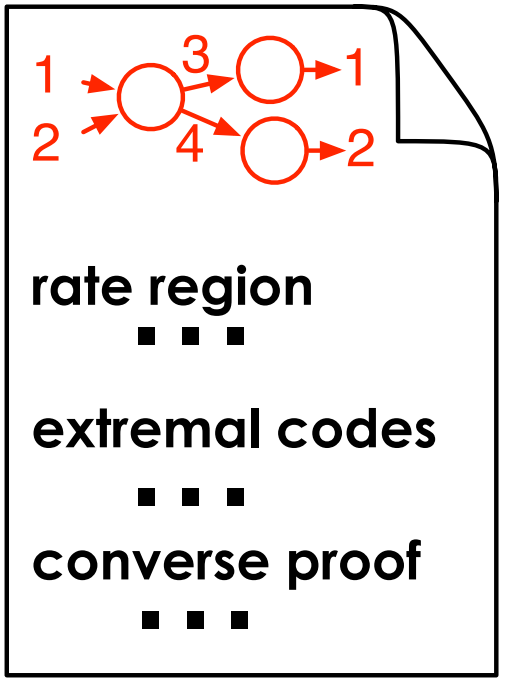
▪ ▪ ▪

Database of ~**7000** Rate Regions of >**100k** Networks (Li Trans IT sub 2014)
Database of ~**744k** Rate Regions of > 7M or ~**2.3T** Minimal Networks
(Li Trans IT sub 2015)

Network Coding Rate Region Database



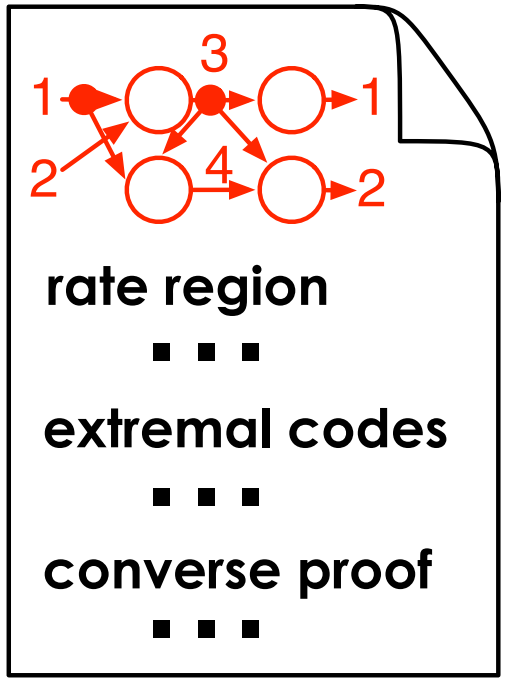


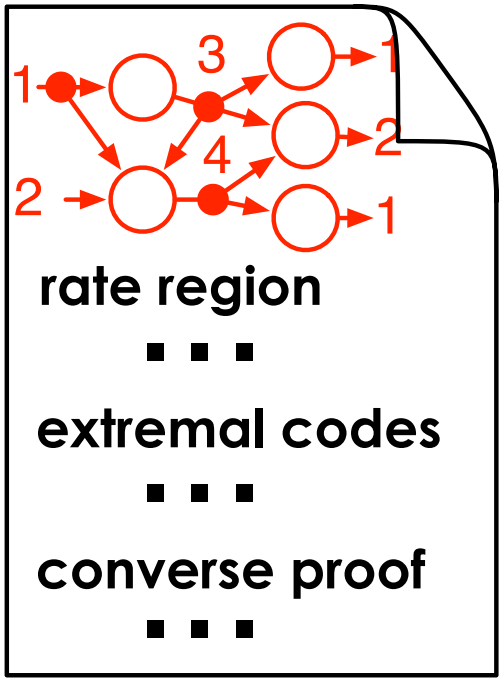


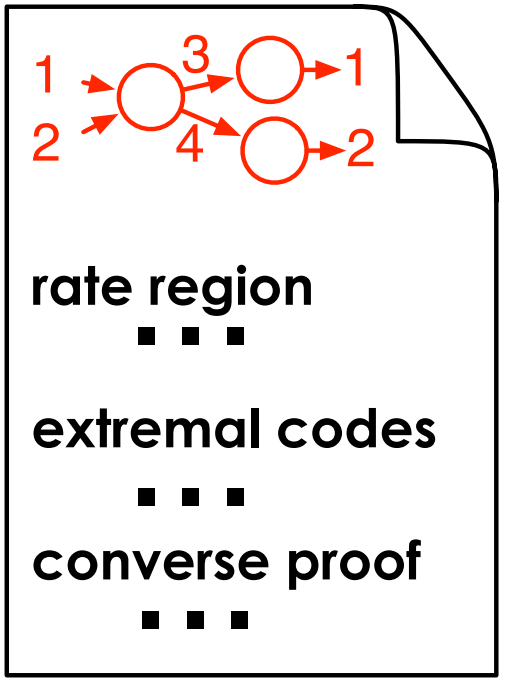
Database of ~**7000** Rate Regions of >**100k** Networks (Li Trans IT sub 2014)
Database of ~**744k** Rate Regions of > 7M or ~**2.3T** Minimal Networks (Li Trans IT sub 2015)

What now?

Network Coding Rate Region Database



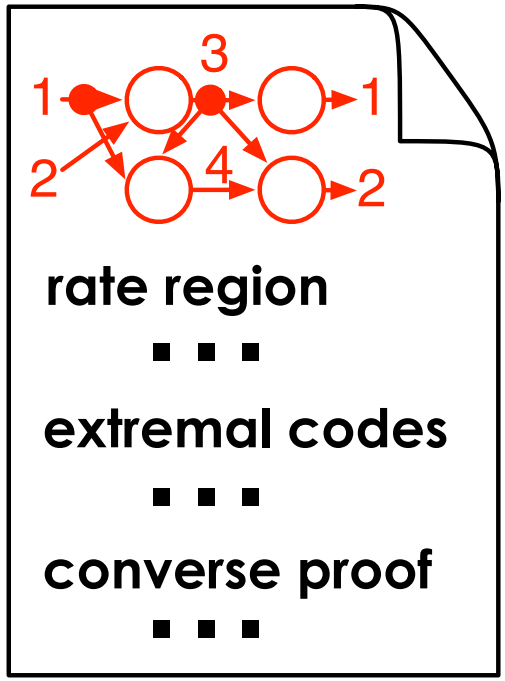




Database of ~**7000** Rate Regions of >**100k** Networks (Li Trans IT sub 2014)
Database of ~**744k** Rate Regions of > 7M or ~**2.3T** Minimal Networks (Li Trans IT sub 2015)

What now? Submit 744,000 transactions papers?

Network Coding Rate Region Database



rate region

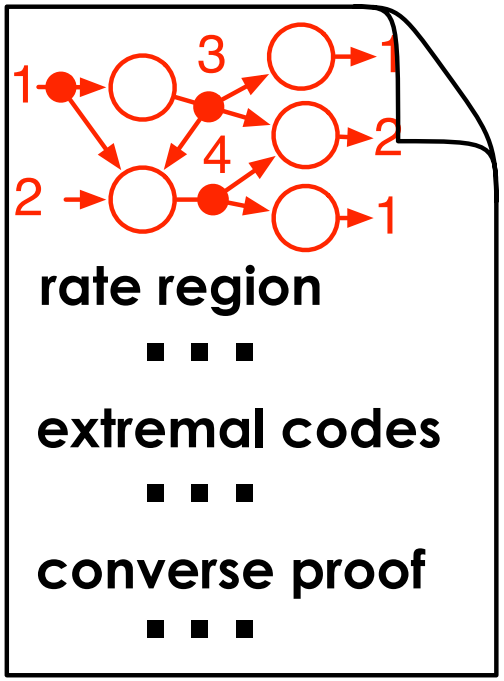
▪ ▪ ▪

extremal codes

▪ ▪ ▪

converse proof

▪ ▪ ▪



rate region

▪ ▪ ▪

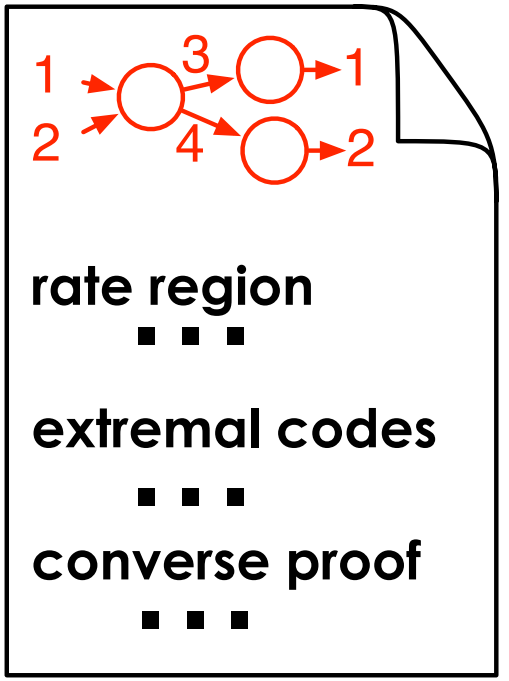
extremal codes

▪ ▪ ▪

converse proof

▪ ▪ ▪

▪ ▪ ▪



rate region

▪ ▪ ▪

extremal codes

▪ ▪ ▪

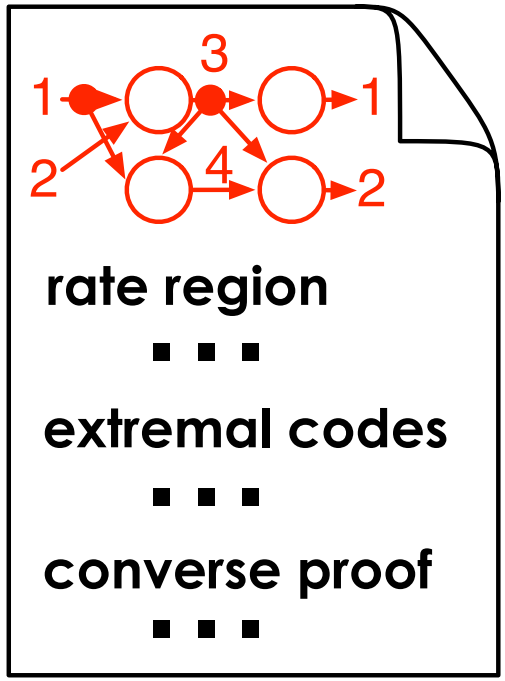
converse proof

▪ ▪ ▪

Database of ~**7000** Rate Regions of >**100k** Networks (Li Trans IT sub 2014)
Database of ~**744k** Rate Regions of > 7M or ~**2.3T** Minimal Networks
(Li Trans IT sub 2015)

What now? Analyze, learn, and explain something!
Can't read and remember 744,000 network proofs.

Network Coding Rate Region Database



rate region

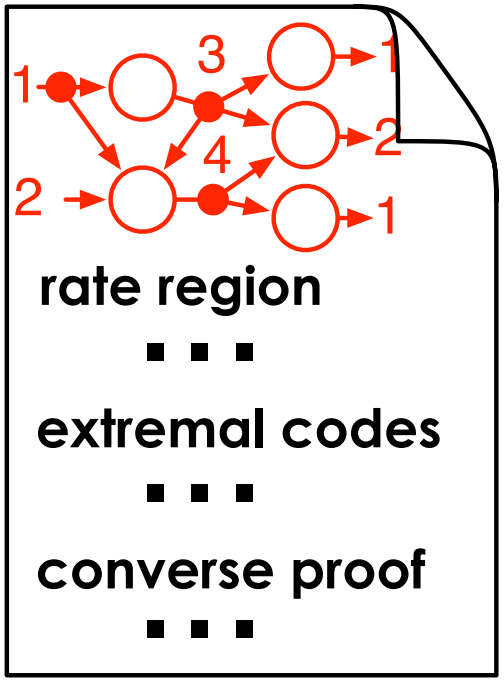
▪ ▪ ▪

extremal codes

▪ ▪ ▪

converse proof

▪ ▪ ▪



rate region

▪ ▪ ▪

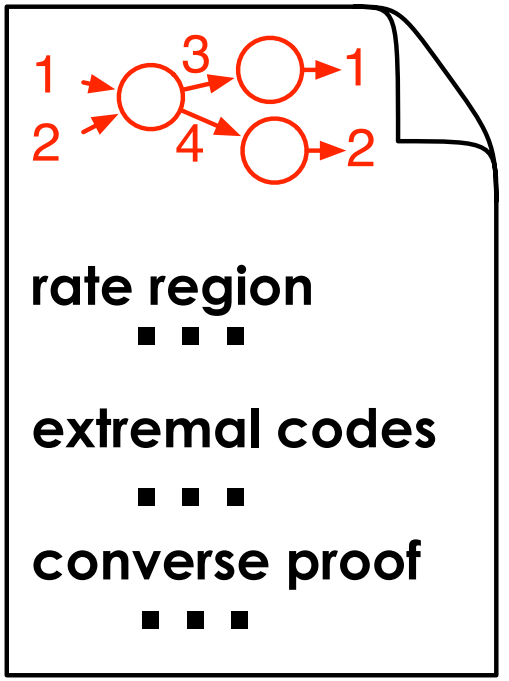
extremal codes

▪ ▪ ▪

converse proof

▪ ▪ ▪

▪ ▪ ▪



rate region

▪ ▪ ▪

extremal codes

▪ ▪ ▪

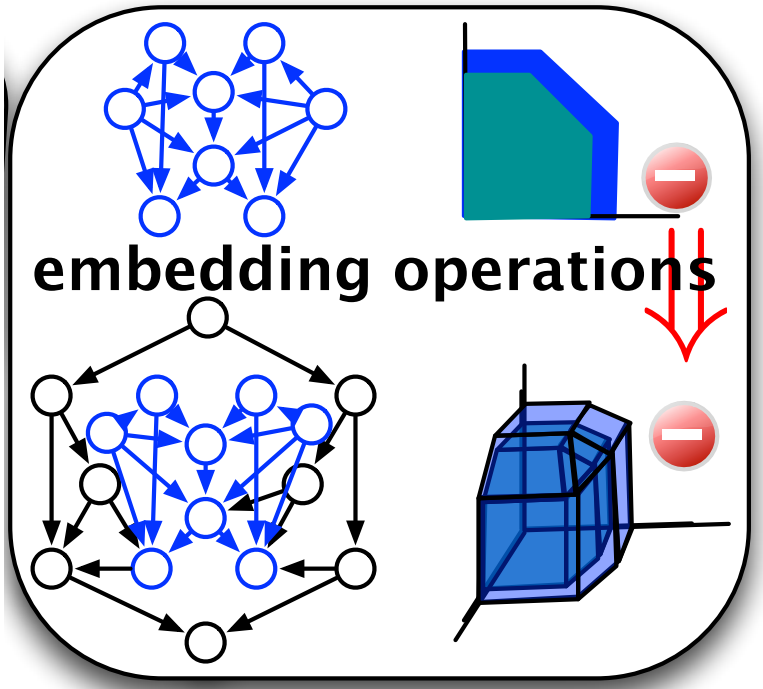
converse proof

▪ ▪ ▪

Database of ~**7000** Rate Regions of >**100k** Networks (Li Trans IT sub 2014)
Database of ~**744k** Rate Regions of > 7M or ~**2.3T** Minimal Networks
(Li Trans IT sub 2015)

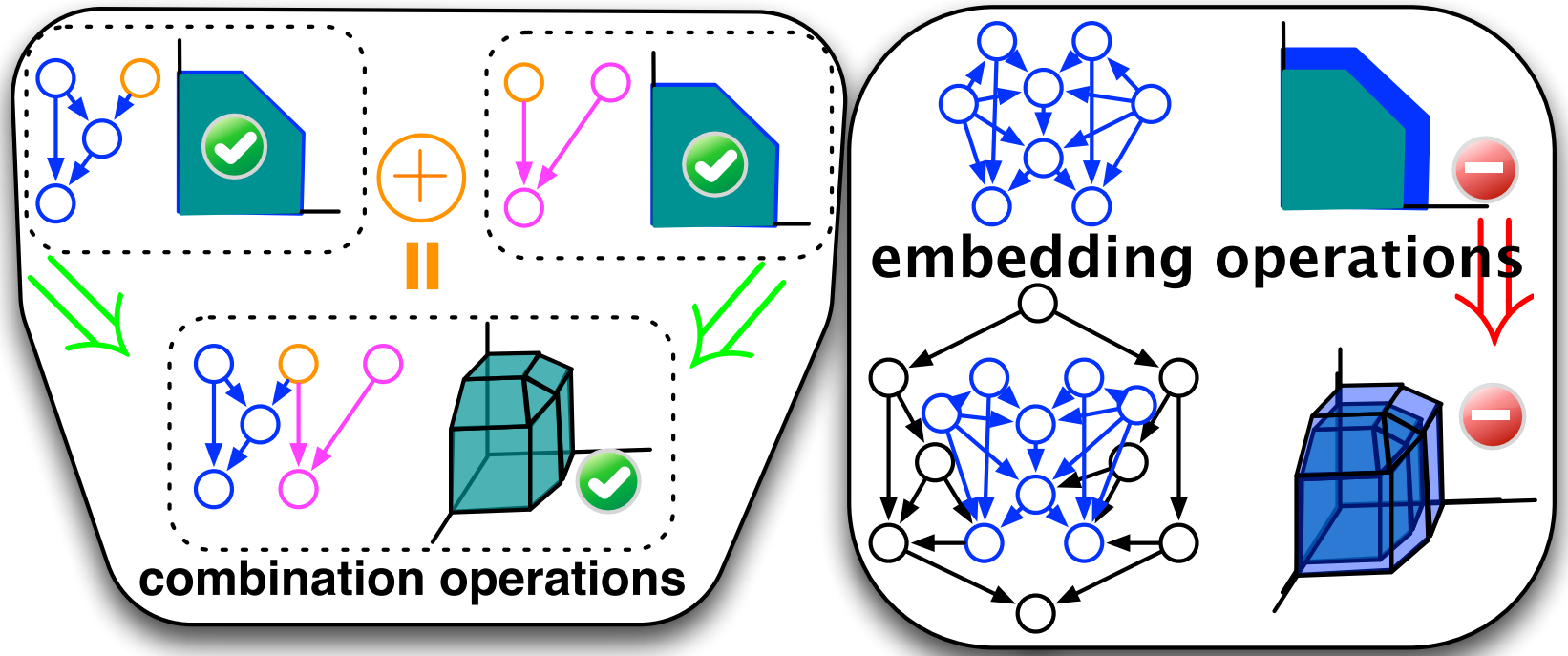
Investigate Structure through Network Hierarchy

Computationally Enabled Research Agenda – Hierarchy



- Understand existing large rate region database through structure: embedding operators & forbidden network minors

Computationally Enabled Research Agenda – Hierarchy

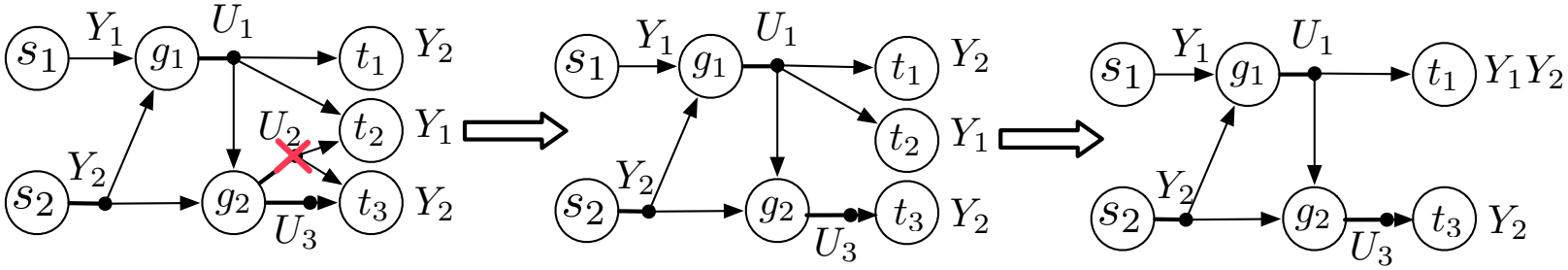


Network well-quasi-ordering Forbidden network Minors

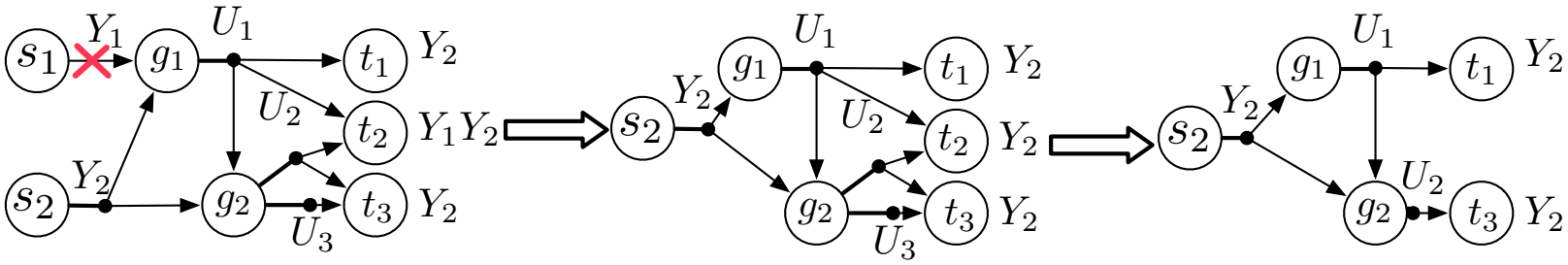
- Understand existing large rate region database through structure: embedding operators & forbidden network minors
- Build solutions to arbitrarily sized networks: combinations operators, operator concatenation

Computationally Enabled Research Agenda – Hierarchy: Embedding Operators

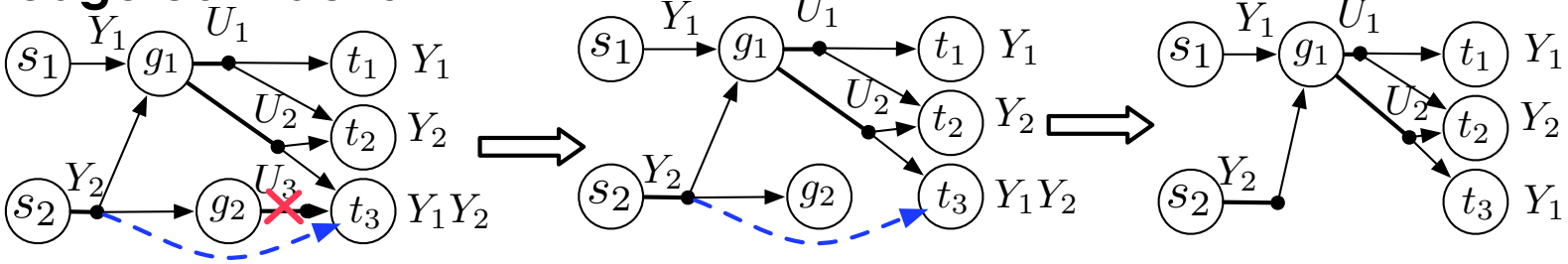
edge deletion



source deletion

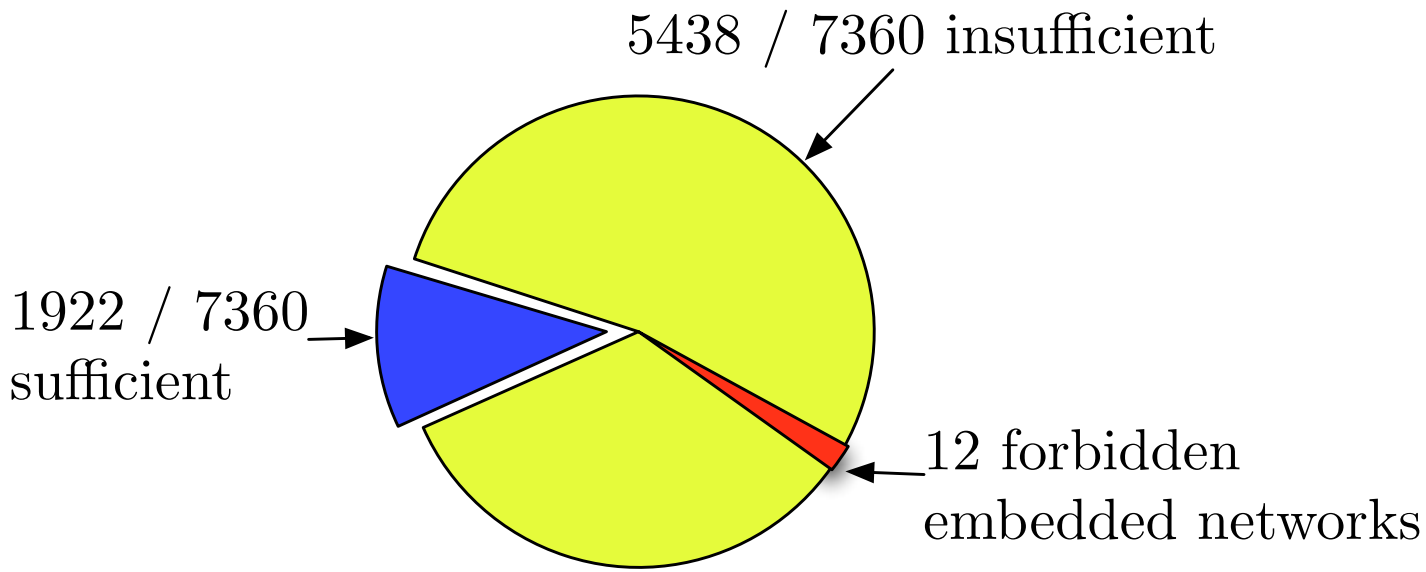


edge contraction



- Rate region (bound) of embedded network can be directly obtained from rate region (bound) of parent network.
- Insufficiency of class of codes of small $\Longrightarrow$ insufficiency of class of codes of big. (*forbidden network minor*)

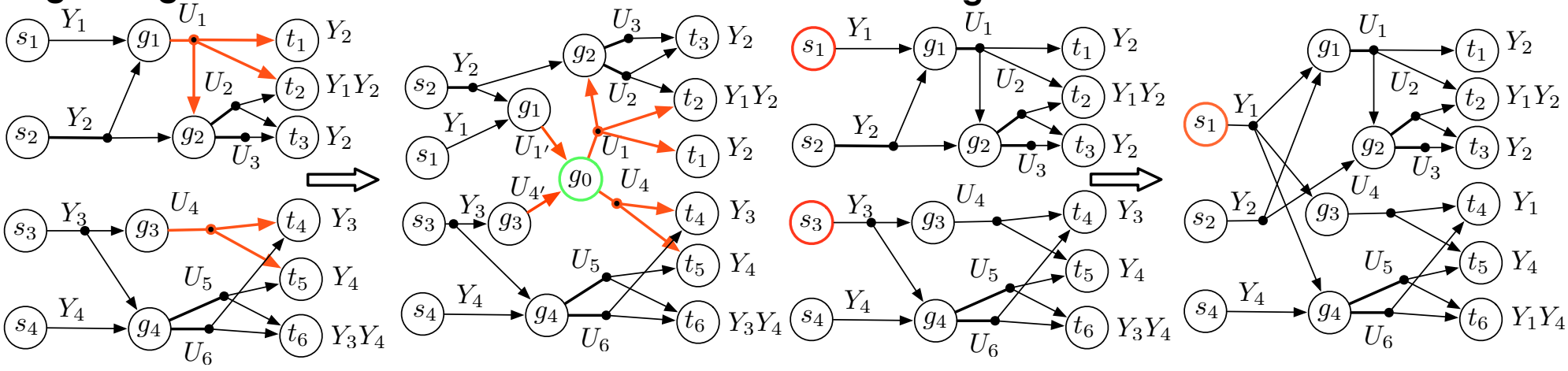
Computationally Enabled Research Agenda – Hierarchy: Embedding Operators



- First database: 5438 canonical MDCS for which scalar binary codes are insufficient can be boiled down to 12 forbidden minor networks.

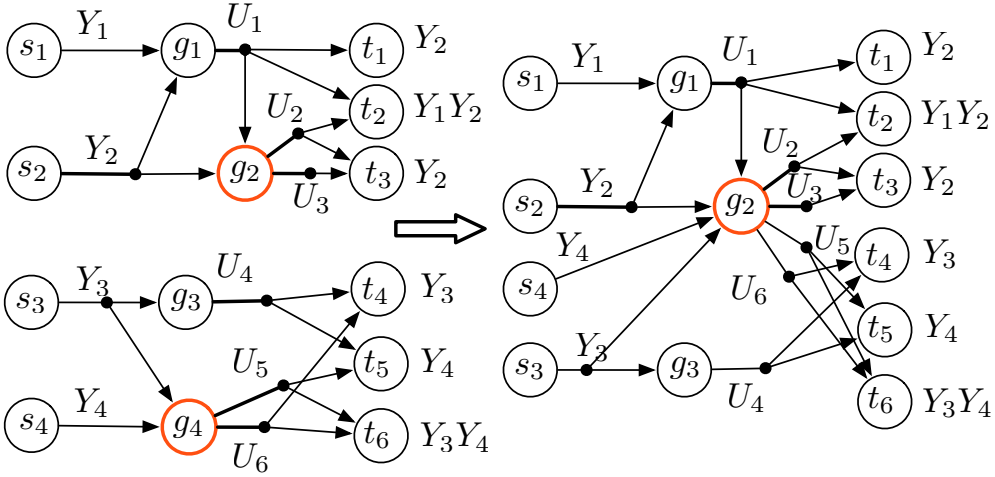
Computationally Enabled Research Agenda – Hierarchy: Combination Operators

Edge Merge

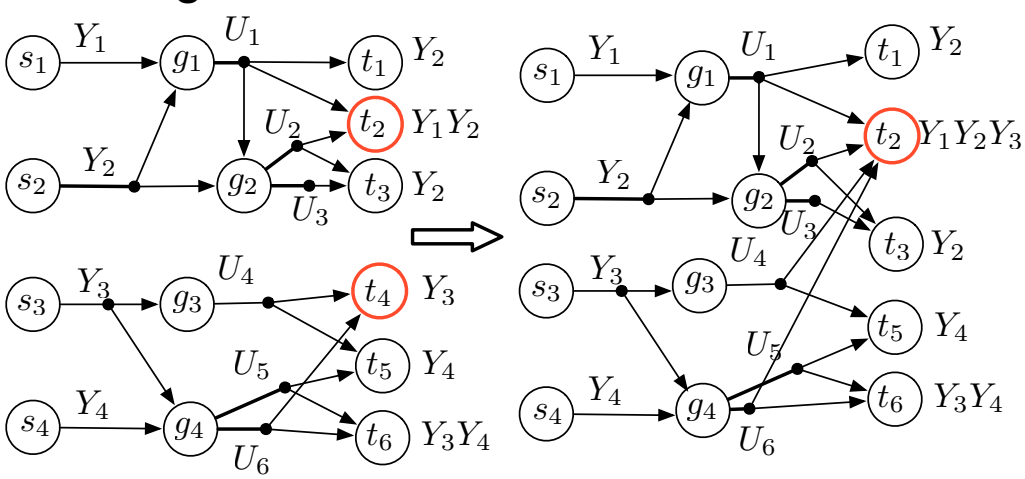


Source Merge

Node Merge

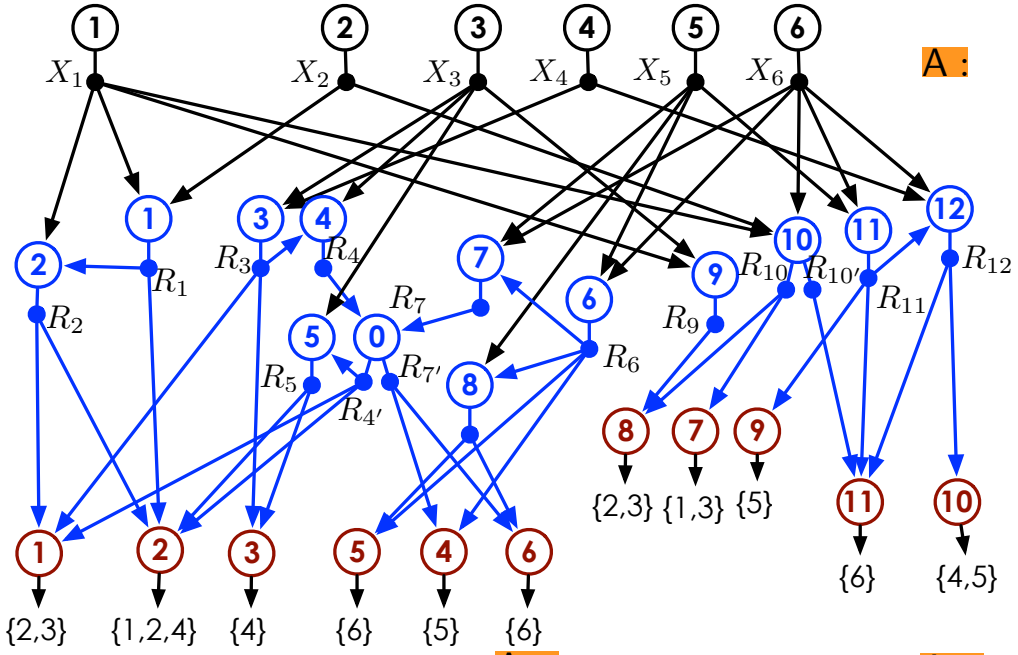


Sink Merge

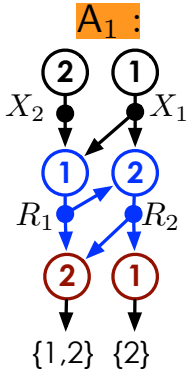


- rate region of big directly expressible from rate regions of smalls

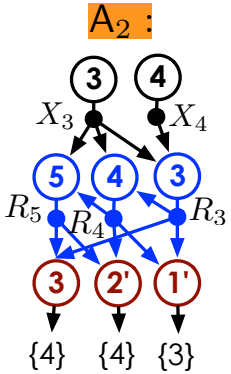
Computationally Enabled Research Agenda – Hierarchy: Combination Operators



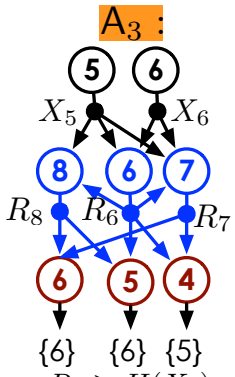
$$
\begin{array}{rcl}
R_i & \geq & H(X_2), i = 1, 2 \\
R_1 + R_2 & \geq & H(X_1) + H(X_2) \\
R_i & \geq & H(X_4), i = 3, 4, 4' \\
R_3 + R_i & \geq & H(X_3) + H(X_4), i = 4, 4' \\
R_3 + R_i + R_5 & \geq & H(X_3) + 2H(X_4), i = 4, 4' \\
R_6 & \geq & H(X_6) \\
R_i + R_8 & \geq & H(X_6), i = 7, 7' \\
R_6 + R_i & \geq & H(X_5) + H(X_6), i = 7, 7' \\
R_6 + R_i + 2R_8 & \geq & H(X_5) + 2H(X_6), i = 7, 7' \\
R_9 & \geq & H(X_3) \\
R_{10} & \geq & \sum_{i=1}^{3} H(X_i) \\
R_{11} & \geq & H(X_5) \\
R_{12} & \geq & H(X_4) + H(X_5) \\
R_{10'} + R_{11} + R_{12} & \geq & H(X_4) + 2H(X_5) + H(X_6)
\end{array}
$$



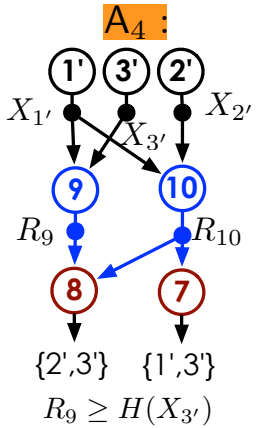
$$R_1 \geq H(X_2)$$
$$R_2 \geq H(X_2)$$
$$R_1 + R_2 \geq H(X_1) + H(X_2)$$



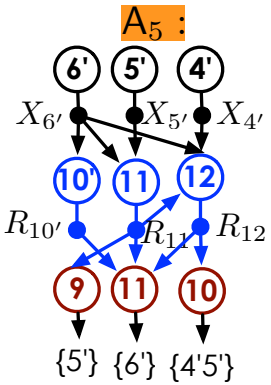
$$R_i \geq H(X_4), i = 3, 4$$
$$R_3 + R_4 \geq H(X_3) + H(X_4)$$
$$R_3 + R_4 + 2R_5 \geq H(X_3) + 2H(X_4)$$



$$R_6 \geq H(X_6)$$
$$R_7 + R_8 \geq H(X_6)$$
$$R_6 + R_7 \geq H(X_5) + H(X_6)$$
$$R_6 + R_7 + 2R_8 \geq H(X_5) + 2H(X_6)$$

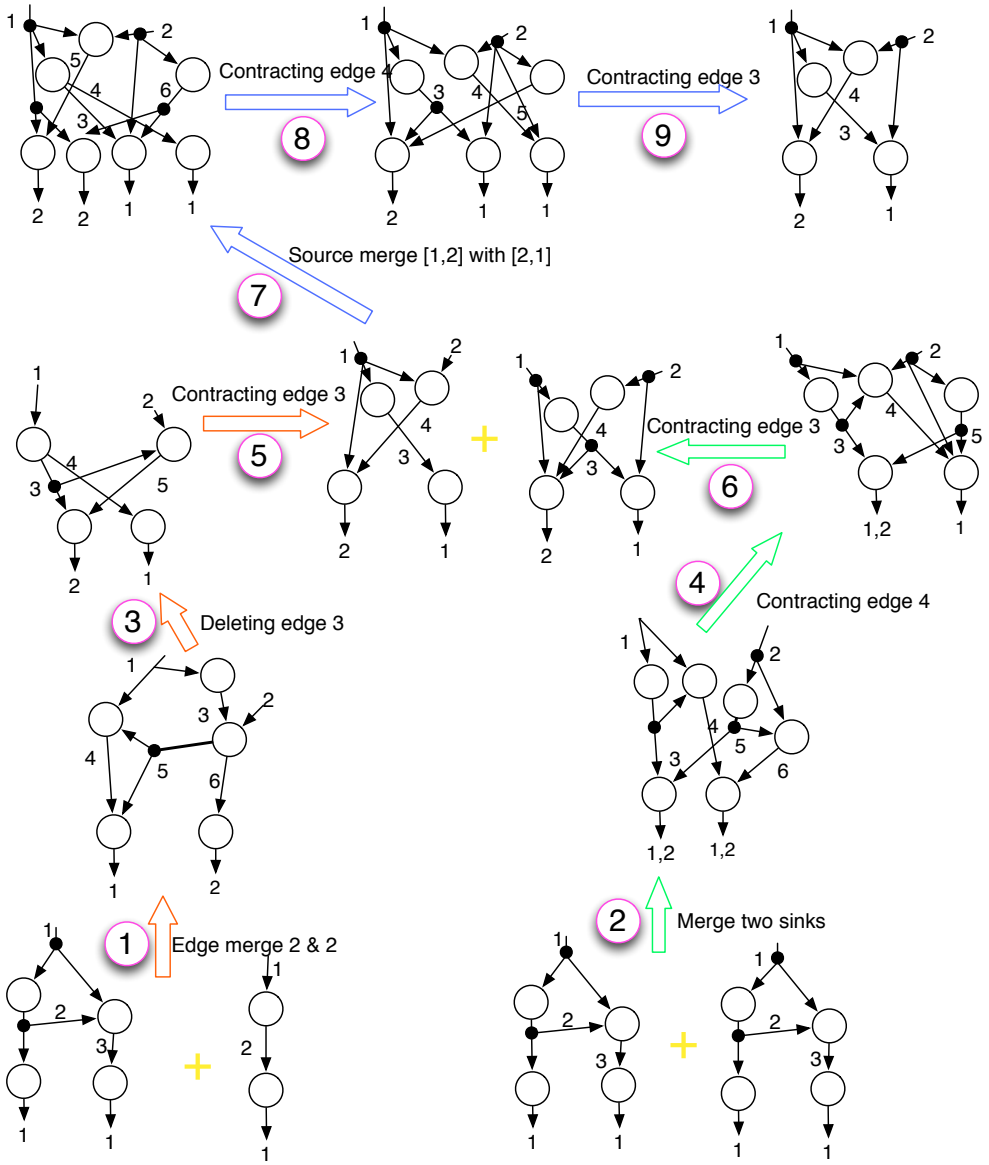


$$R_9 \geq H(X_{3'})$$
$$R_{10} \geq H(X_{1'}) + H(X_{2'}) + H(X_{3'})$$

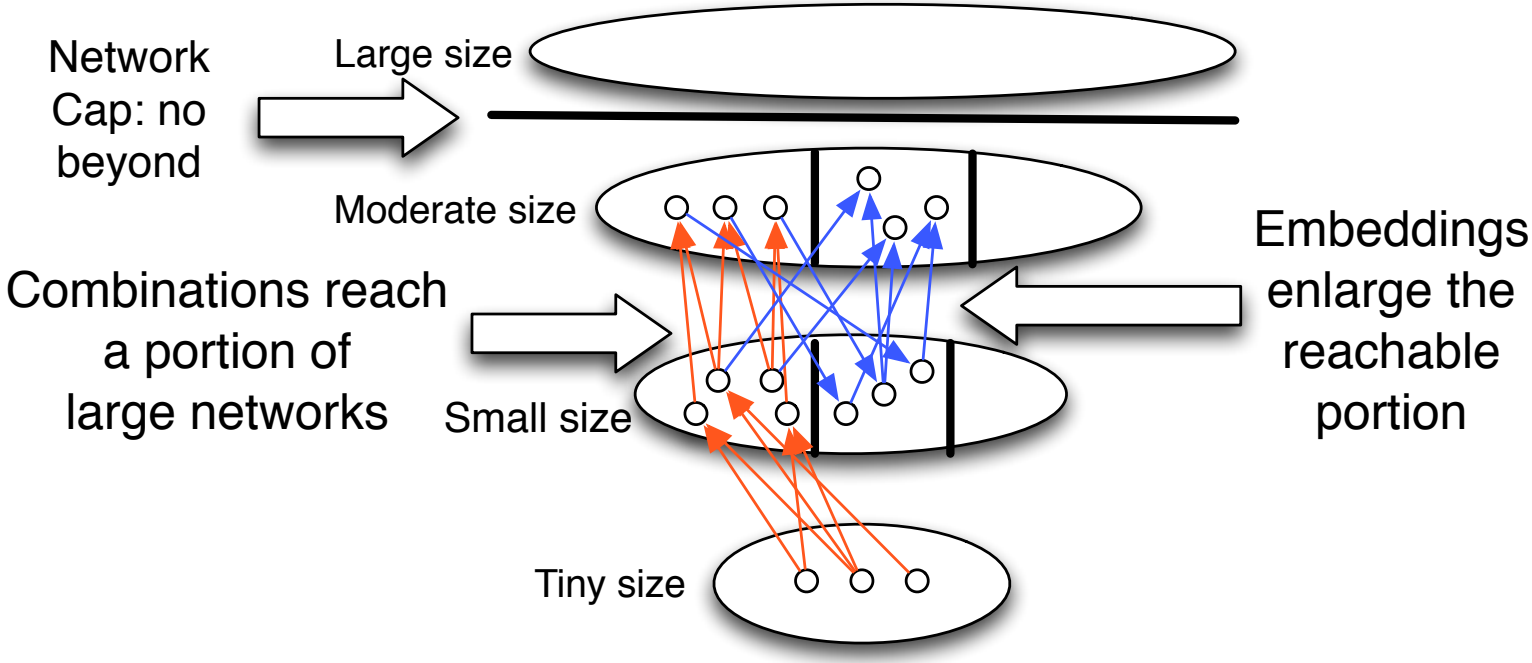


$$R_{11} \geq H(X_{5'})$$
$$R_{12} \geq H(X_{4'}) + H(X_{5'})$$
$$R_{10'} + R_{11} + R_{12} \geq H(X_{4'}) + 2H(X_{5'}) + H(X_{6'})$$

Computationally Enabled Research Agenda – Hierarchy: Operator Concatenation



Computationally Enabled Research Agenda – Hierarchy: Operator Concatenation



Use operators *together* to get RR for big networks. Partial Network Closure.

Computationally Enabled Research Agenda – Hierarchy: Operator Concatenation

Start with the single $(1,1)$, single $(2,1)$, and the four $(1,2)$ networks; These 6 tiny networks can generate new 11635 networks w/ small cap!

	combination operators only			embedding and combinations		
size\cap	(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

Computationally Enabled Research Agenda – Hierarchy: Operator Concatenation

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

	combination operators only			embedding and combinations		
size\cap	(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

Computationally Enabled Research Agenda – Hierarchy: Operator Concatenation

Embedding operations are important in the process!

	combination operators only			embedding and combinations		
size\cap	(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