Parameterized Complexity of Almost-Satisfiable Constraint Systems

George Osipov

Royal Holloway, University of London

Parameterized Complexity of Computational Reasoning
(PCCR 2026)

Almost-Satisfiable Constraint Systems

Based on joint work with

- Konrad K Dabrowski (Newcastle University, UK)
- Peter Jonsson (Linköping University, Sweden)
- Sebastian Ordyniak (University of Leeds, UK)
- Marcin Pilipczuk (University of Warsaw, Poland)
- Roohani Sharma (Institute for Basic Science, Korea)
- Jorke de Vlas (Linköping University, Sweden)
- Magnus Wahlström (Royal Holloway, University of London, UK)

Almost-Satisfiable Constraint Systems

Example: Crime Investigation

Witness 1: “I heard the alarm and the window glass shatter at the same time.”
Witness 2: “The guard called after the window was broken.”
Witness 3: “The guard called before the alarm.”

Almost-Satisfiable Constraint Systems

Example: Crime Investigation

Witness 1: “I heard the **a**larm and the **w**indow glass shatter *at the same time*.”
Witness 2: “The **g**uard called *after* the **w**indow was broken.”
Witness 3: “The **g**uard called *before* the **a**larm.”

Almost-Satisfiable Constraint Systems

Example: Crime Investigation

Witness 1: “I heard the **a**larm and the **w**indow glass shatter *at the same time*.”
Witness 2: “The **g**uard called *after* the **w**indow was broken.”
Witness 3: “The **g**uard called *before* the **a**larm.”

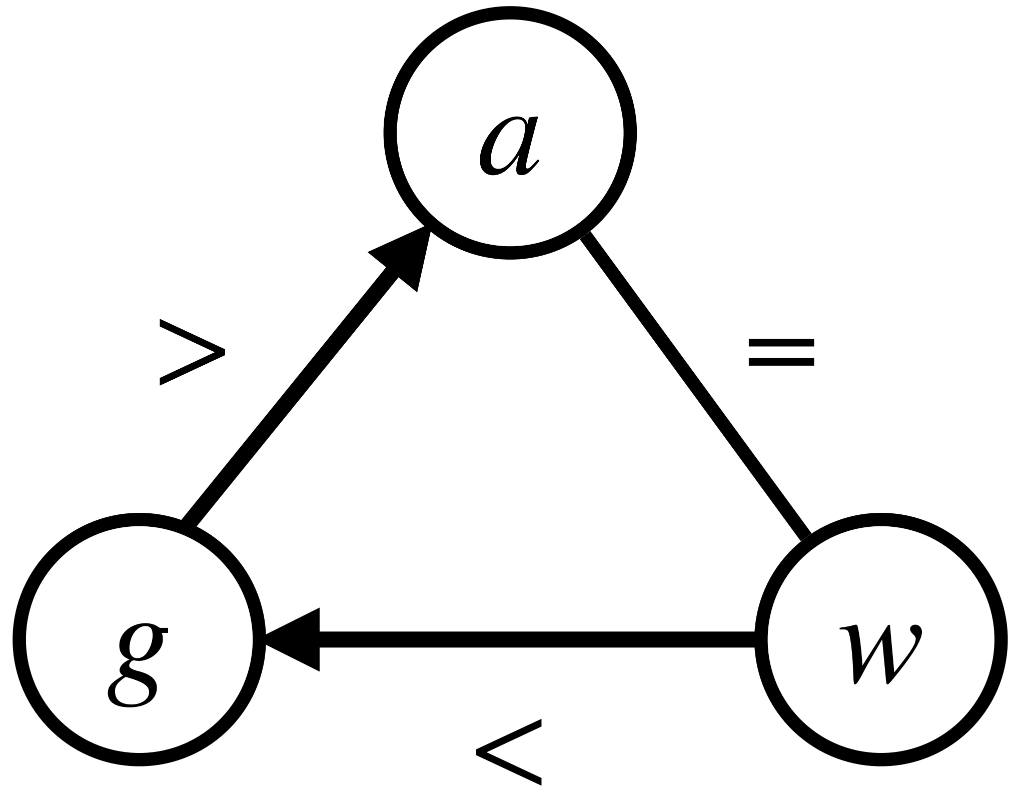
Witness 1: $a = w$
Witness 2: $g > w$
Witness 3: $g < a$

Almost-Satisfiable Constraint Systems

Example: Crime Investigation

Witness 1: “I heard the **a**larm and the **w**indow glass shatter *at the same time*.”
Witness 2: “The **g**uard called *after* the **w**indow was broken.”
Witness 3: “The **g**uard called *before* the **a**larm.”

Witness 1: $a = w$
Witness 2: $g > w$
Witness 3: $g < a$



The evidence is inconsistent.
How many witnesses are giving false information?

Parameterized Optimization Problems

Problem	Input	Deleting	Goal
Vertex Cover	A graph and an integer k	k vertices	no edges
Bipartization **OCT**	A graph and an integer k	k edges k vertices	bipartite
DFAS	A directed graph and an integer k	k arcs	acyclic
Multiway Cut	A graph with some terminal vertices, and an integer k	k edges k vertices	terminals disconnected
Hitting Set	A hypergraph and an integer k	k vertices	no hyperedges

Parameterized Optimization Problems

Problem	Input	Deleting	Goal	Complexity
Vertex Cover	A graph and k	k vertices	no edges	FPT
Bipartization **OCT**	A graph and k	k edges k vertices	bipartite	FPT
DFAS	A directed graph and k	k arcs	acyclic	FPT
Multiway Cut	A graph with some terminal vertices, and k	k edges k vertices	terminals disconnected	FPT
Hitting Set	A hypergraph and k	k vertices	no hyperedges	W[1]-hard

Parameterized Optimization Problems

Problem	Input	Deleting	Goal	Complexity
Vertex Cover	A graph and k	k vertices	no edges	FPT
Bipartization **OCT**	A graph and k	k edges k vertices	bipartite	FPT
DFAS	A directed graph and k	k arcs	acyclic	FPT
Multiway Cut	A graph with some terminal vertices, and k	k edges k vertices	terminals disconnected	FPT
Hitting Set	A hypergraph and k	k vertices	no hyperedges	W[1]-hard

These problems (and many more) are special cases of MinCSP.

Plan

1. Gentle introduction to CSPs (followed by a quiz)
2. Point Algebra: Advertisement
3. Boolean MinCSP and Flow Augmentation
4. Algorithms for Point Algebra
5. Some open problems

Gentle Introduction to CSPs

Constraint Satisfaction Problem (CSP)

CSP over domain D.

Input: (V, C), where V is a set of variables and C is a collection of constraints.

Goal: an assignment $\varphi : V \to D$ satisfying all constraints in C.

Each constraint is a pair (R, X), where

- $X = (x_1, \ldots, x_r) \in V^r$ is a list of r variables called the scope, and
- $R \subseteq D^r$ is the set of allowed assignments to X called a constraint relation.
- φ satisfies (R, X) if $(\varphi(x_1), \ldots, \varphi(x_r)) \in R$.

Constraint Satisfaction Problem (CSP)

Example 1

Domain $D = \{1,2,3\}$

Variables: $V = \{a, b, c, d\}$

Constraints:

- $(\{11,22,33\}, (a, b))$, equivalently $(a = b)$
- $(\{12,13,23\}, (b, c))$, equivalently $(b < c)$
- $(\{2,3\}, d)$, equivalently $(d \neq 1)$
- $(\{12,23\}, (c, d))$, equivalently $(c + 1 = d)$

Constraint Satisfaction Problem (CSP)

Example 1

Domain $D = \{1,2,3\}$

Variables: $V = \{a, b, c, d\}$

Constraints:

- $(\{11,22,33\}, (a, b))$, equivalently $(a = b)$
- $(\{12,13,23\}, (b, c))$, equivalently $(b < c)$
- $(\{2,3\}, d)$, equivalently $(d \neq 1)$
- $(\{12,23\}, (c, d))$, equivalently $(c + 1 = d)$

Satisfying assignment $\varphi(a, b, c, d) = (1,1,2,3)$

Constraint Satisfaction Problem (CSP)

Example 1

Domain $D = \{1,2,3\}$

Variables: $V = \{a, b, c, d\}$

Constraints:

- $(\{\mathbf{11},22,33\}, (a, b))$, equivalently $(a = b)$
- $(\{\mathbf{12},13,23\}, (b, c))$, equivalently $(b < c)$
- $(\{\mathbf{2},3\}, d)$, equivalently $(d \neq 1)$
- $(\{12,\mathbf{23}\}, (c, d))$, equivalently $(c + 1 = d)$

Satisfying assignment $\varphi(a, b, c, d) = (1,1,2,3)$

Constraint Satisfaction Problem (CSP)

Example 2: k-Coloring

Domain $D = \{1,\ldots,k\}$

Variables: vertices of the graph

Constraints: $(u \neq v)$ for every edge uv of the graph

Constraint Satisfaction Problem (CSP)

Example 3: Digraph Acyclicity

Domain $D = \mathbb{Q}$

Variables: vertices of the digraph

Constraints: $(u < v)$ for every arc (u, v) of the digraph

Constraint Satisfaction Problem (CSP)

Non-Uniform Setting

CSP(Γ) over domain D and constraint language Γ

Input: (V, C), where V is a set of variables and C is a collection of constraints.

Goal: an assignment $\alpha : V \to D$ satisfying all constraints in C.

Each constraint is a pair (R, X), where R belongs to Γ.

- $D = \{1,2,3\}, \Gamma = \{ \neq \}$ is 3-Coloring
- $D = \mathbb{Q}, \Gamma = \{ < \}$ is Digraph Acyclicity

Minimum-Cost CSP (MinCSP)

MinCSP(Γ) over domain D and constraint language Γ

Input: (V, C, k), where (V, C) is an instance of CSP(Γ)

Parameter: k

Goal: an assignment $\alpha : V \to D$ of cost at most k, where cost = # unsatisfied constraints

Minimum-Cost CSP (MinCSP)

Deletion Formulation

MinCSP(Γ) over domain D and constraint language Γ

Input: (V, C, k), where (V, C) is an instance of CSP(Γ)

Parameter: k

Goal: is there $X \subseteq C$, $|X| \leq k$, such that $(V, C \backslash X)$ is satisfiable?

Constraint Satisfaction Problem (CSP)

Examples with Graph Problems

Domain	Constraints	CSP	MinCSP
{0, 1}	$x \neq y$	?	?
{1, 2, 3}	$x \neq y$	?	?
{0, 1}	$x=1$ or $y=1$, $x=0$	?	?
Q	$x < y$	?	?
N	$x = y$, $x=1$, $x=2$, …	?	?

Constraint Satisfaction Problem (CSP)

Examples with Graph Problems

Cheat Sheet:

Problem: deleting … / goal

Vertex Cover: vertices / no edges

Bipartization: edges / bipartite

DFAS: arcs / no cycles

Multiway Cut: edges / no terminal paths

Domain	Constraints	CSP	MinCSP
{0, 1}	$x \neq y$	?	?
{1, 2, 3}	$x \neq y$	?	?
{0, 1}	$x=1$ or $y=1$, $x=0$	?	?
Q	$x < y$	?	?
N	$x = y$, $x=1$, $x=2$, …	?	?

Constraint Satisfaction Problem (CSP)

Examples with Graph Problems

Cheat Sheet:

Problem: deleting … / goal

Vertex Cover: vertices / no edges

Bipartization: edges / bipartite

DFAS: arcs / no cycles

Multiway Cut: edges / no terminal paths

Domain	Constraints	CSP	MinCSP
{0, 1}	x ≠ y	2-coloring	?
{1, 2, 3}	x ≠ y	?	?
{0, 1}	x=1 or y=1, x=0	?	?
Q	x < y	?	?
N	x = y, x=1, x=2, …	?	?

Constraint Satisfaction Problem (CSP)

Examples with Graph Problems

Cheat Sheet:

Problem: deleting … / goal

Vertex Cover: vertices / no edges

Bipartization: edges / bipartite

DFAS: arcs / no cycles

Multiway Cut: edges / no terminal paths

Domain	Constraints	CSP	MinCSP
{0, 1}	$x \neq y$	2-coloring	Bipartization
{1, 2, 3}	$x \neq y$	?	?
{0, 1}	$x=1$ or $y=1$, $x=0$	?	?
Q	$x < y$	?	?
N	$x = y$, $x=1$, $x=2$, …	?	?

Constraint Satisfaction Problem (CSP)

Examples with Graph Problems

Cheat Sheet:

Problem: deleting … / goal

Vertex Cover: vertices / no edges

Bipartization: edges / bipartite

DFAS: arcs / no cycles

Multiway Cut: edges / no terminal paths

Domain	Constraints	CSP	MinCSP
{0, 1}	$x \neq y$	2-coloring	Bipartization
{1, 2, 3}	$x \neq y$	3-coloring	NP-h for k=0
{0, 1}	$x=1$ or $y=1$, $x=0$	?	?
Q	$x < y$	?	?
N	$x = y$, $x=1$, $x=2$, …	?	?

Constraint Satisfaction Problem (CSP)

Examples with Graph Problems

Cheat Sheet:

Problem: deleting … / goal

Vertex Cover: vertices / no edges

Bipartization: edges / bipartite

DFAS: arcs / no cycles

Multiway Cut: edges / no terminal paths

Domain	Constraints	CSP	MinCSP
{0, 1}	$x \neq y$	2-coloring	Bipartization
{1, 2, 3}	$x \neq y$	3-coloring	NP-h for k=0
{0, 1}	$x=1$ or $y=1$, $x=0$	no edge is 0-0	?
Q	$x < y$	?	?
N	$x = y$, $x=1$, $x=2$, …	?	?

Constraint Satisfaction Problem (CSP)

Examples with Graph Problems

Cheat Sheet:

Problem: deleting … / goal

Vertex Cover: vertices / no edges

Bipartization: edges / bipartite

DFAS: arcs / no cycles

Multiway Cut: edges / no terminal paths

Domain	Constraints	CSP	MinCSP
{0, 1}	$x \neq y$	2-coloring	Bipartization
{1, 2, 3}	$x \neq y$	3-coloring	NP-h for k=0
{0, 1}	x=1 or y=1, x=0	no edge is 0-0	Vertex Cover*
Q	$x < y$	?	?
N	x = y, x=1, x=2, …	?	?

Constraint Satisfaction Problem (CSP)

Examples with Graph Problems

Cheat Sheet:

Problem: deleting … / goal

Vertex Cover: vertices / no edges

Bipartization: edges / bipartite

DFAS: arcs / no cycles

Multiway Cut: edges / no terminal paths

Domain	Constraints	CSP	MinCSP
{0, 1}	$x \neq y$	2-coloring	Bipartization
{1, 2, 3}	$x \neq y$	3-coloring	NP-h for k=0
{0, 1}	x=1 or y=1, x=0	no edge is 0-0	Vertex Cover*
Q	$x < y$	no cycles	?
N	x = y, x=1, x=2, …	?	?

Constraint Satisfaction Problem (CSP)

Examples with Graph Problems

Cheat Sheet:

Problem: deleting … / goal

Vertex Cover: vertices / no edges

Bipartization: edges / bipartite

DFAS: arcs / no cycles

Multiway Cut: edges / no terminal paths

Domain	Constraints	CSP	MinCSP
{0, 1}	$x \neq y$	2-coloring	Bipartization
{1, 2, 3}	$x \neq y$	3-coloring	NP-h for k=0
{0, 1}	x=1 or y=1, x=0	no edge is 0-0	Vertex Cover*
Q	$x < y$	no cycles	DFAS
N	x = y, x=1, x=2, …	?	?

Constraint Satisfaction Problem (CSP)

Examples with Graph Problems

Cheat Sheet:

Problem: deleting … / goal

Vertex Cover: vertices / no edges

Bipartization: edges / bipartite

DFAS: arcs / no cycles

Multiway Cut: edges / no terminal paths

Domain	Constraints	CSP	MinCSP
{0, 1}	$x \neq y$	2-coloring	Bipartization
{1, 2, 3}	$x \neq y$	3-coloring	NP-h for k=0
{0, 1}	x=1 or y=1, x=0	no edge is 0-0	Vertex Cover*
Q	$x < y$	no cycles	DFAS
N	x = y, x=1, x=2, …	no path i=…=j	?

Constraint Satisfaction Problem (CSP)

Examples with Graph Problems

Cheat Sheet:

Problem: deleting … / goal

Vertex Cover: vertices / no edges

Bipartization: edges / bipartite

DFAS: arcs / no cycles

Multiway Cut: edges / no terminal paths

Domain	Constraints	CSP	MinCSP
{0, 1}	$x \neq y$	2-coloring	Bipartization
{1, 2, 3}	$x \neq y$	3-coloring	NP-h for k=0
{0, 1}	x=1 or y=1, x=0	no edge is 0-0	Vertex Cover*
Q	$x < y$	no cycles	DFAS
N	x = y, x=1, x=2, …	no path i=…=j	Multiway Cut

Extra Credit

For Roohani

Domain - subsets of a 3-element set: $\varnothing, \{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}$

Unary relations: $1 \in X$, $2 \in X$, $3 \in X$, $1 \notin X$, $2 \notin X$, $3 \notin X$

Binary relation: $X \subseteq Y$

What is this MinCSP?

Extra Credit

For Roohani

Domain - subsets of a 3-element set: $\varnothing, \{1\}, \{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}, \{1,2,3\}$

Unary relations: $1 \in X$, $2 \in X$, $3 \in X$, $1 \notin X$, $2 \notin X$, $3 \notin X$

Binary relation: $X \subseteq Y$

What is this MinCSP? Directed Multicut with Three Terminal Pairs

Dichotomy Theorems

XP vs paraNP-hard

Theorem [Bulatov 2017, Zhuk 2017]:
Let Γ be a constraint language over a finite domain.
Then CSP(Γ) is either in P or NP-hard.

Observe: MinCSP(Γ) with $k = 0$ is precisely CSP(Γ).

Corollary:
Let Γ be a constraint language over a finite domain.
Then MinCSP(Γ) is either paraNP-hard or in XP. // easy $n^{O(k)}$ algorithm

When is MinCSP(Γ) in **FPT**?

Dichotomy Theorems

FPT vs W[1]-hard

Dream: For every "nice" Γ, MinCSP(Γ) is either in FPT or W[1]-hard.

"nice" $\supseteq$ finite-domain, FO-definable from $(\mathbb{N}; =)$ or $(\mathbb{Q}; <)$, …

Progress so far:

- Boolean MinCSP (2-element domain): [KKPW 2023]
- Equality MinCSP (FO-definable from $(\mathbb{N}; =)$): [OW 2023]

Dichotomy Theorems

FPT vs W[1]-hard

Other projects in this direction:

- List H-Coloring by Removing Few Vertices [CEM ESA'2013]
- O(1)-approximation for Boolean MinCSP [BELM ESA’2016]
- Interval Algebra [DJOOPS IPEC'2023]
- Linear Equations over $\mathbb{Q}$, $\mathbb{Z}$ [DJOOW SODA’2023], $\mathbb{Z}_m$ [–, ESA’2025 + FOCS’2026]
- Point Algebra [OPW ESA’2024]
- DTPs [DJOOdV KR’2026]

Dichotomy Theorems

FPT vs W[1]-hard

Other projects in this direction:

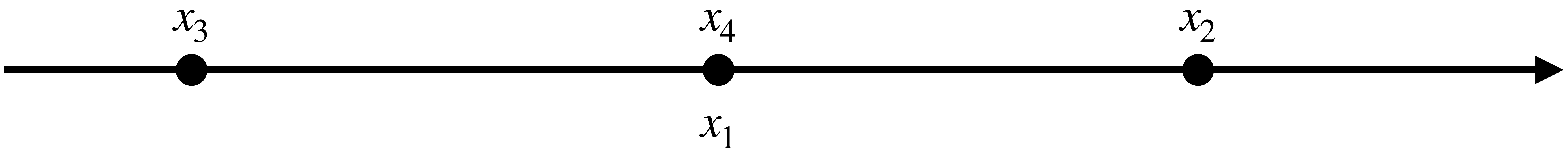
- List H-Coloring by Removing Few Vertices [CEM ESA'2013]
- O(1)-approximation for Boolean MinCSP [BELM ESA’2016]
- Interval Algebra [DJOOPS IPEC'2023]
- Linear Equations over $\mathbb{Q}$, $\mathbb{Z}$ [DJOOW SODA’2023], $\mathbb{Z}_m$ [–, ESA’2025 + FOCS’2026]
- **Point Algebra** [OPW ESA’2024]
- DTPs [DJOOdV KR’2026]

Point Algebra

Advertisement

Point Algebra

Objects: points on the timeline $x_1, x_2, \ldots, x_n$



Domain: $\mathcal{Q}$

Relations: subsets Γ of $<, >, \leq, \geq, =, \neq$

What is the complexity of MinCSP(Γ) for all such Γ?

Point Algebra

Domain: $\mathbb{Q}$

Relations: subsets Γ of $<, >, \leq, \geq, =, \neq$

What is the complexity of MinCSP(Γ) for all such Γ?

Point Algebra

Domain: $\mathbb{Q}$

Relations: subsets Γ of $<, \leq, =, \neq$

What is the complexity of MinCSP(Γ) for all such Γ?

Point Algebra

Domain: $\mathbb{Q}$

Relations: subsets Γ of $<, \leq, =, \neq$

What is the complexity of MinCSP(Γ) for all such Γ?

- MinCSP($=, \neq$) is Edge Multicut **FPT**
- MinCSP($<$) is Directed Feedback Arc Set (DFAS) **FPT**
- MinCSP($<, \leq$) is Subset DFAS **FPT**
- MinCSP($\leq, \neq$) is Directed Symmetric Multicut **W[1]-hard**

Point Algebra

Domain: $\mathbb{Q}$

Relations: subsets Γ of $<, \leq, =, \neq$

What is the complexity of MinCSP(Γ) for all such Γ?

- MinCSP($=, \neq$) is Edge Multicut **FPT** [MR || BDT STOC'2011]
- MinCSP($<$) is Directed Feedback Arc Set (DFAS) **FPT** [CLLRO'S STOC'2008]
- MinCSP($<, \leq$) is Subset DFAS **FPT** [CCHM ICALP'2012]

All FPT cases can be solved using Boolean MinCSP!

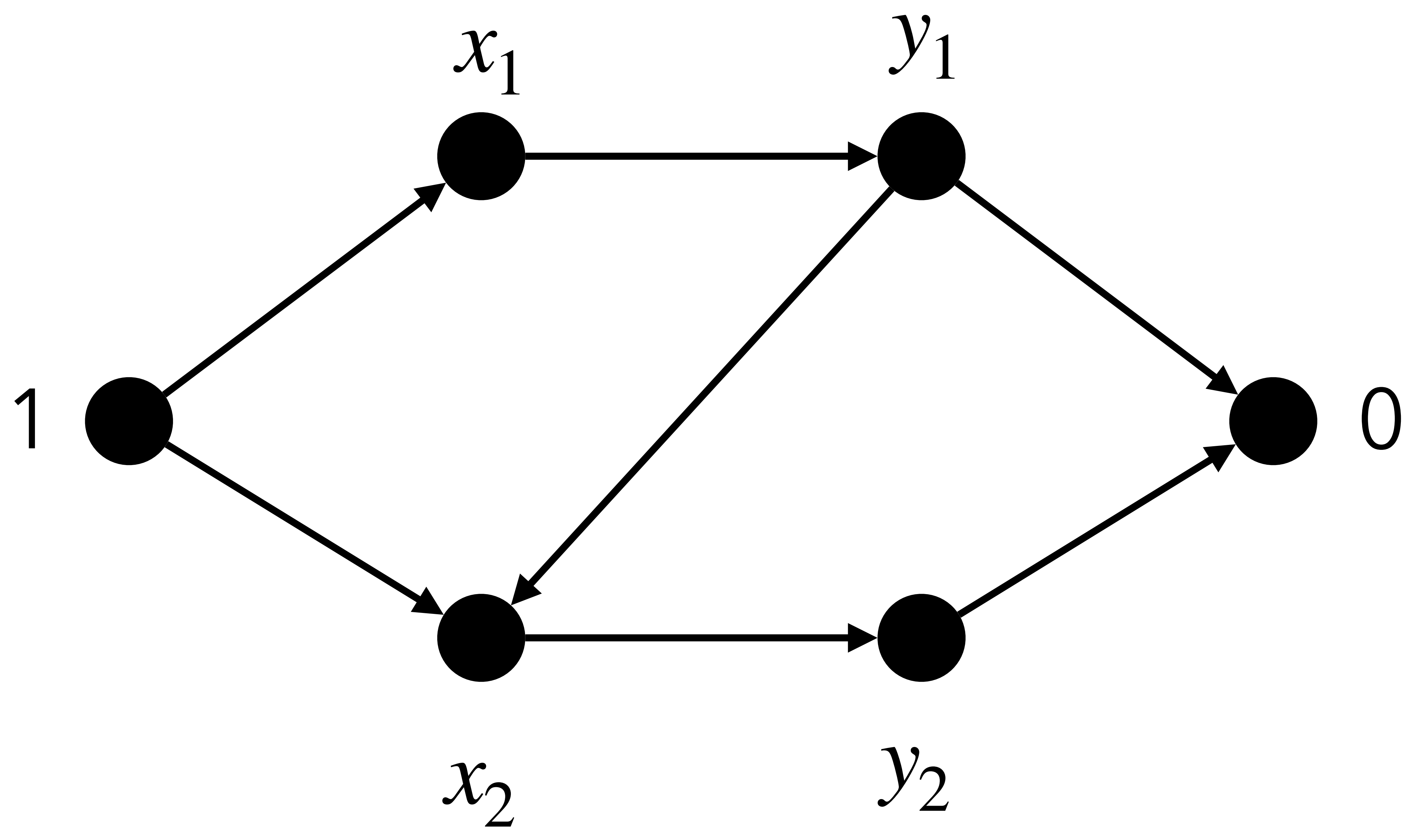
Boolean MinCSP

Based on “Applications of Flow-Augmentation” survey by
S. Kratsch, M. Pilipczuk, R. Sharma and M. Wahlström

Boolean MinCSP and Cuts

MinCSP($\{0,1\}; x = 0,\ x = 1,\ x \to y$) is Minimum st-Cut

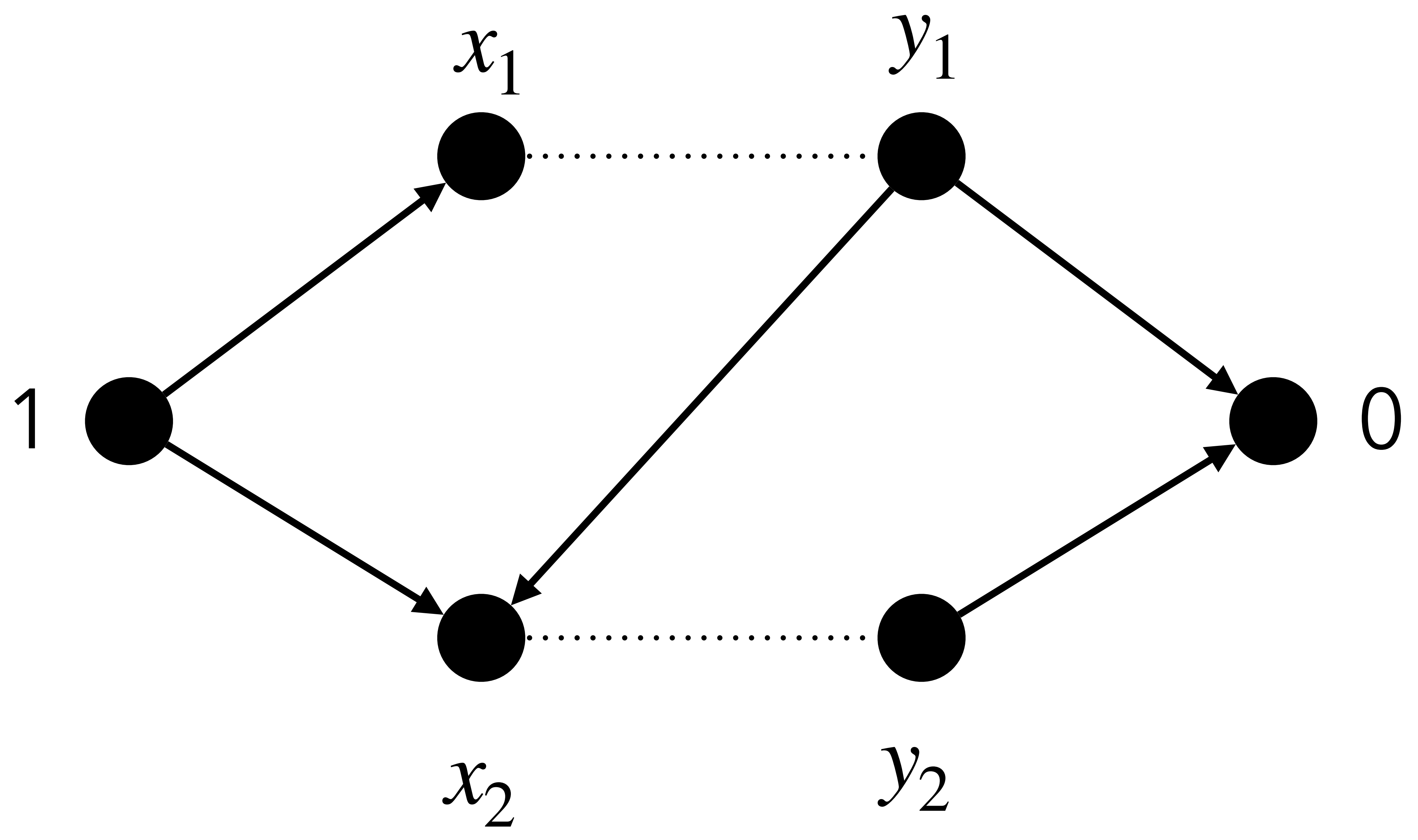
$x_1 = 1$
$x_2 = 1$
$x_1 \to y_1$
$x_2 \to y_2$
$y_1 \to x_2$
$y_1 = 0$
$y_2 = 0$



Boolean MinCSP and Cuts

MinCSP($\{0,1\}; x = 0,\ x = 1,\ x \to y$) is Minimum st-Cut

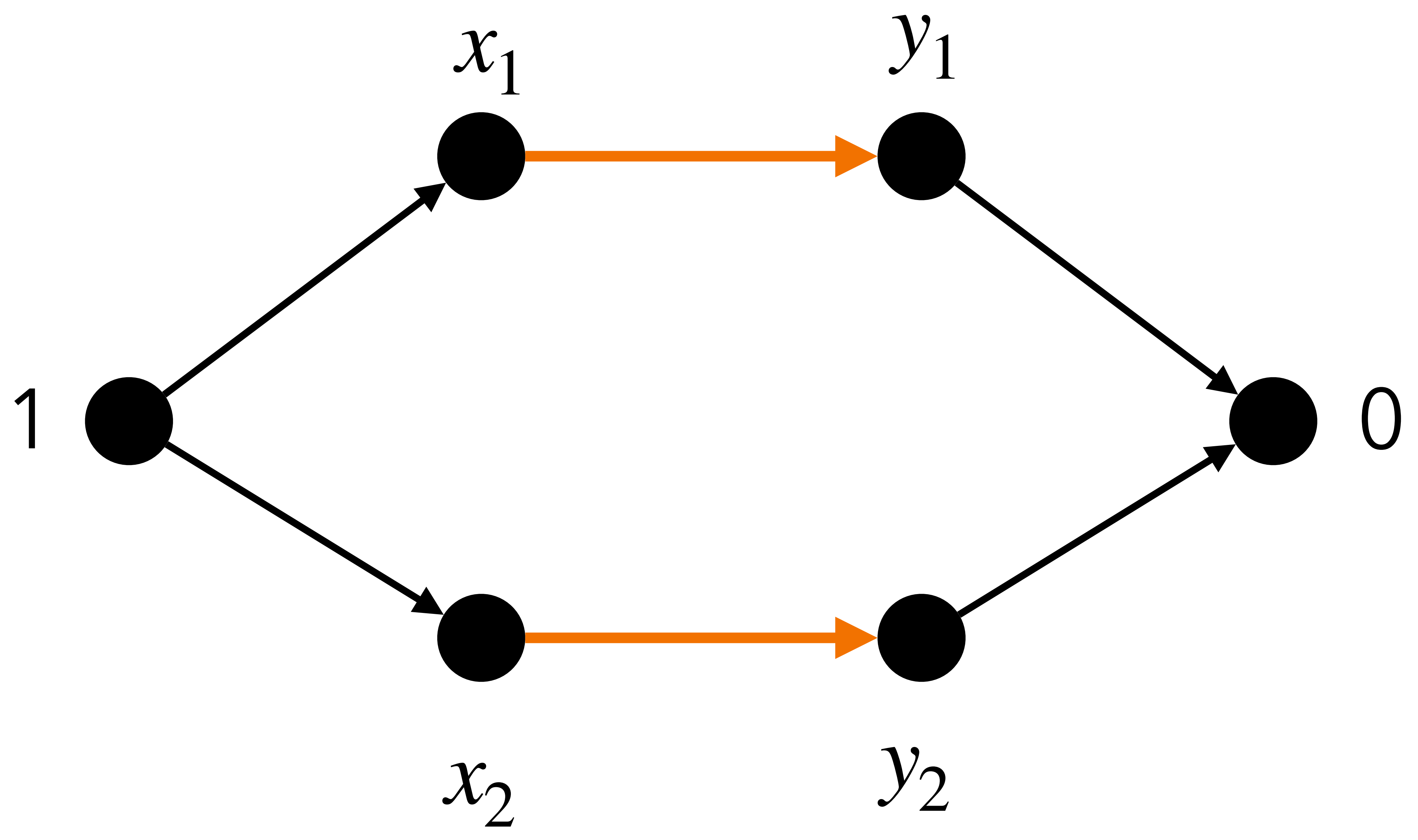
$x_1 = 1$
$x_2 = 1$
$\cancel{x_1 \to y_1}$
$\cancel{x_2 \to y_2}$
$y_1 \to x_2$
$y_1 = 0$
$y_2 = 0$



Boolean MinCSP and Cuts

MinCSP($\{0,1\}; x = 0,\ x = 1,\ (x_1 \to y_1) \wedge (x_2 \to y_2)$) is called Paired MinCut

$x_1 = 1$
$x_2 = 1$
$x_1 \to y_1 \wedge x_2 \to y_2$
$y_1 = 0$
$y_2 = 0$



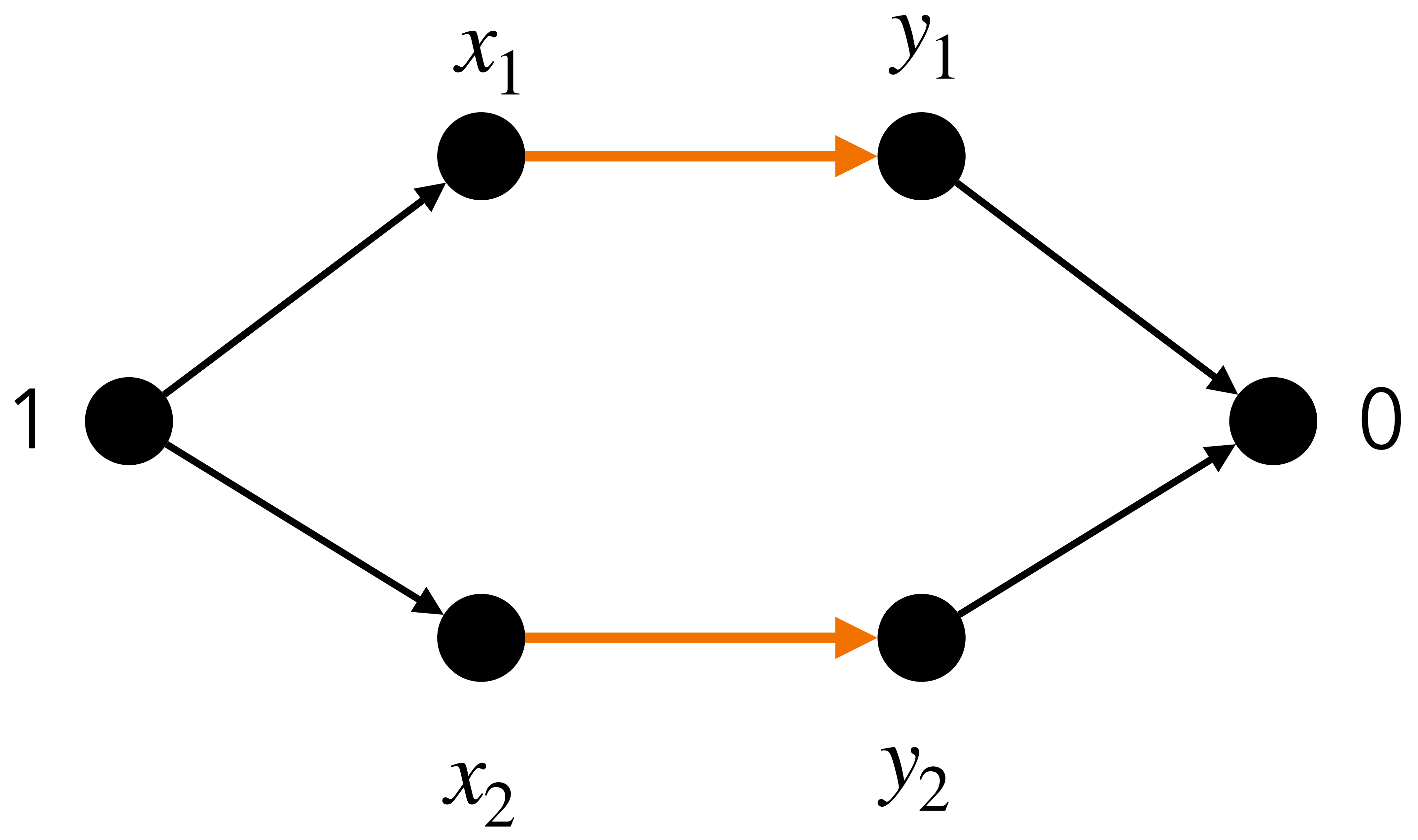
NP-hard. Is it FPT?

Boolean MinCSP and Cuts

MinCSP$(\{0,1\}; x = 0,\ x = 1,\ (x_1 \rightarrow y_1) \wedge (x_2 \rightarrow y_2))$ is called Paired MinCut

$x_1 = 1$
$x_2 = 1$
$x_1 \rightarrow y_1 \wedge x_2 \rightarrow y_2$
$y_1 = 0$
$y_2 = 0$

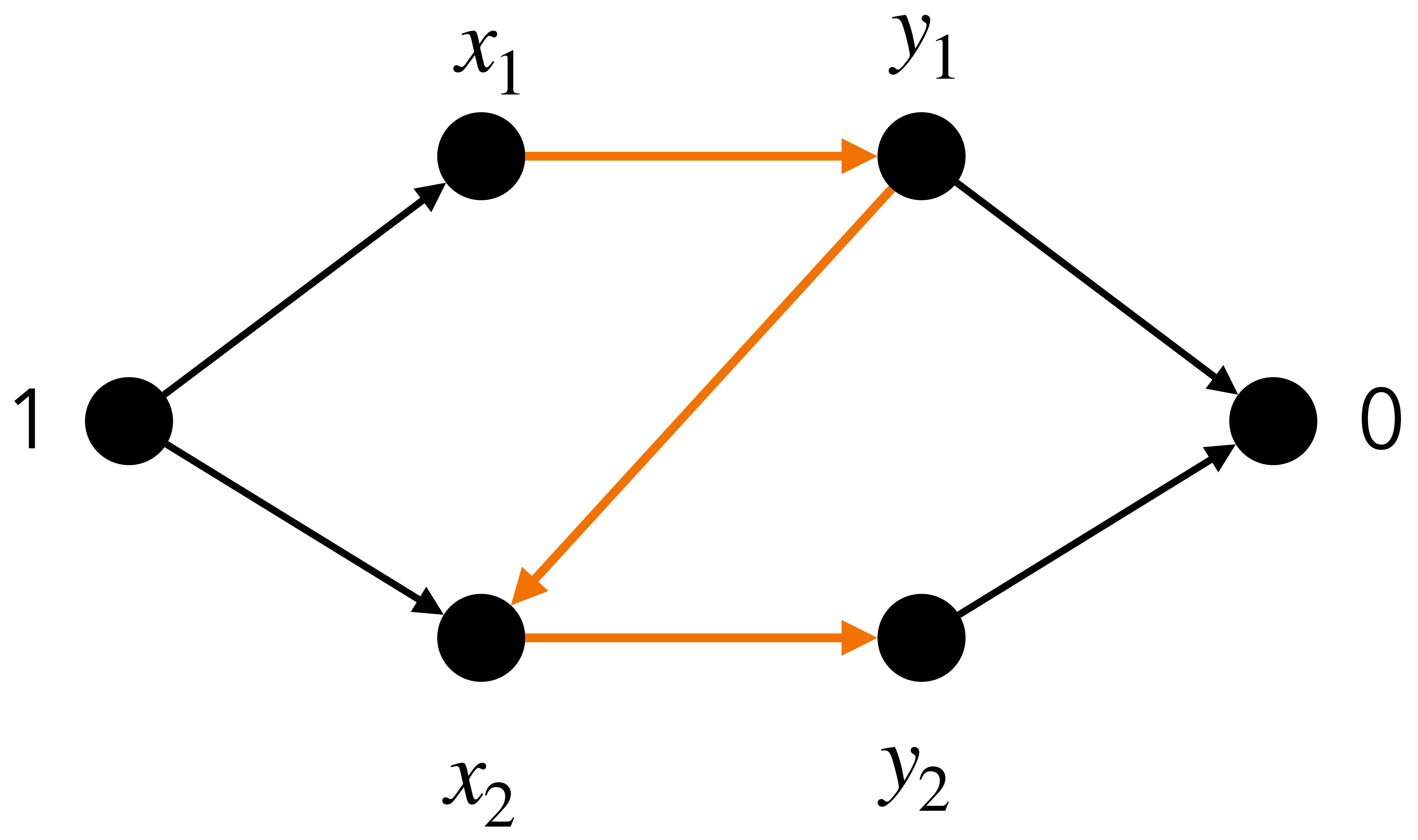
NP-hard. Is it FPT? No, W[1]-hard.



Boolean MinCSP and Cuts

MinCSP($\{0,1\}; x = 0,\ x = 1,\ x_1 \to x_2 \to x_3 \to x_4$) is called 3-Chain SAT

$x_1 = 1$
$x_2 = 1$
$x_1 \to y_1 \to x_2 \to y_2$
$y_1 = 0$
$y_2 = 0$



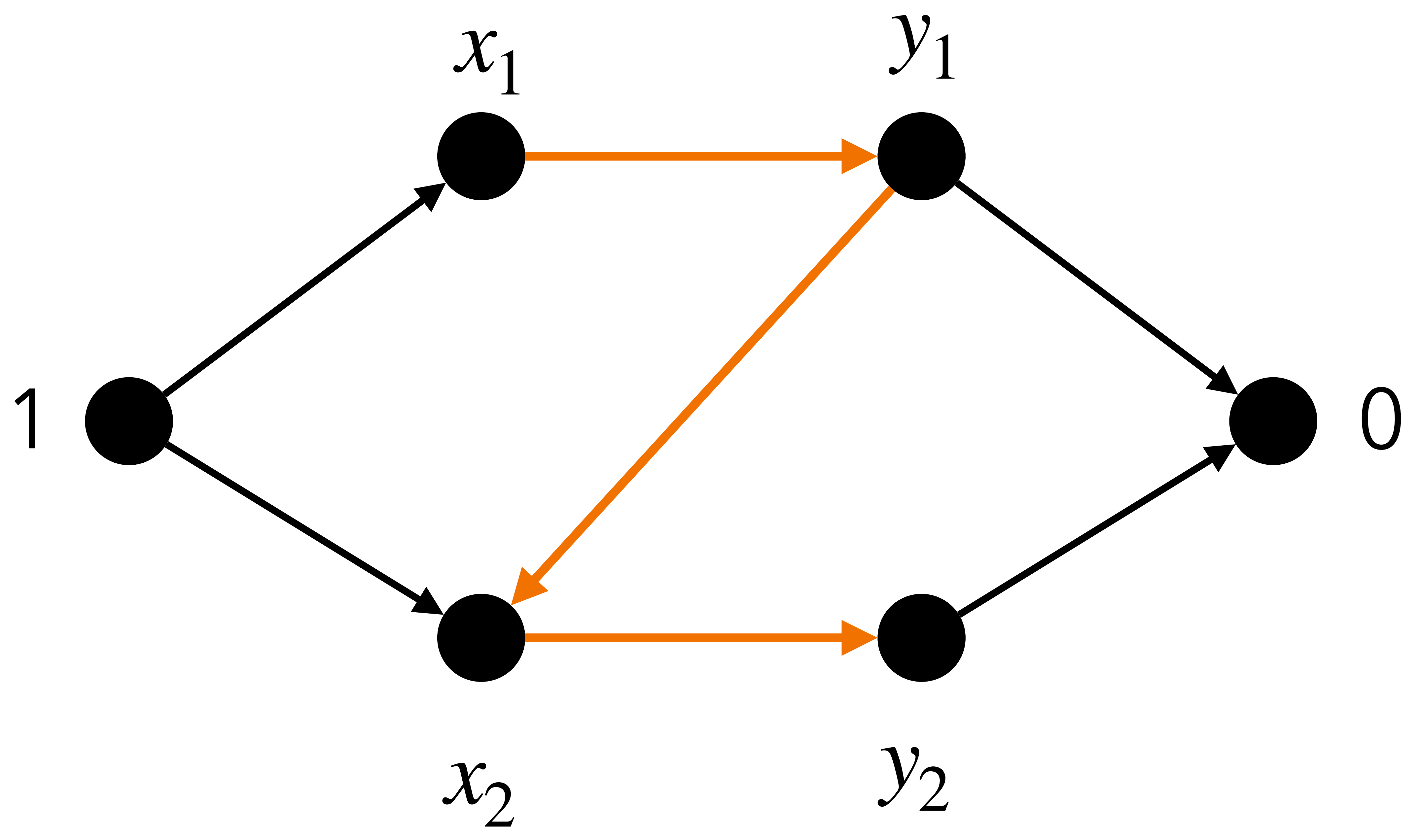
NP-hard. Is it FPT?

Boolean MinCSP and Cuts

MinCSP($\{0,1\}; x = 0,\ x = 1,\ x_1 \to x_2 \to x_3 \to x_4$) is called 3-Chain SAT

$x_1 = 1$
$x_2 = 1$
$x_1 \to y_1 \to x_2 \to y_2$
$y_1 = 0$
$y_2 = 0$

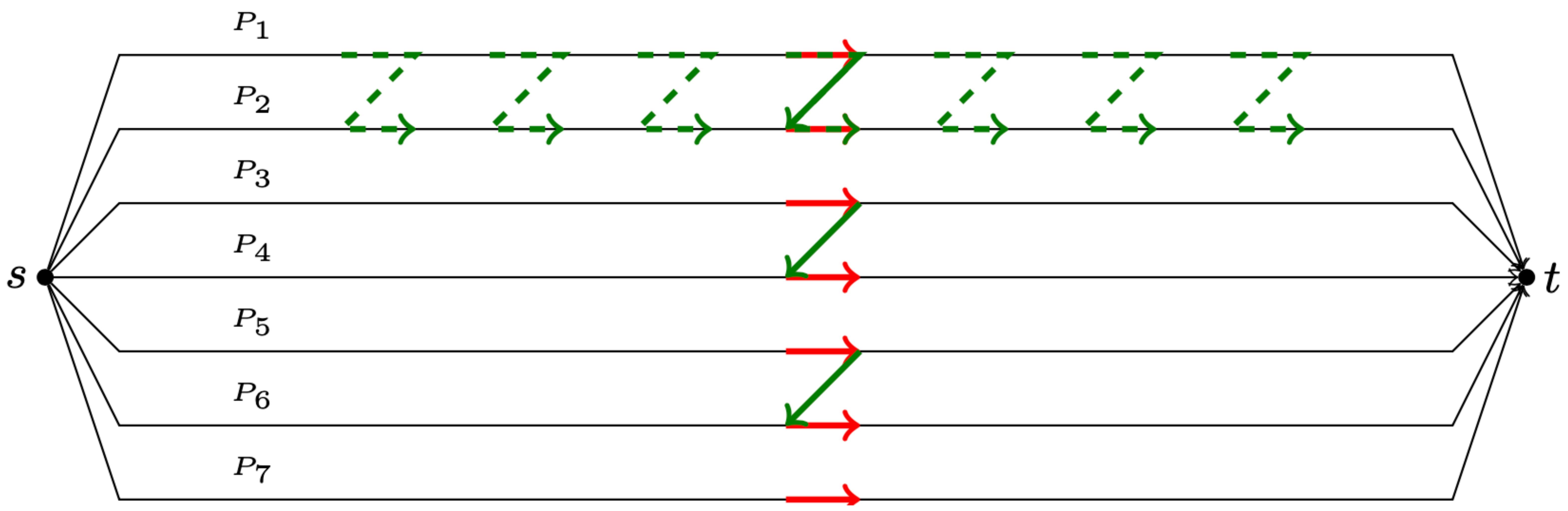
NP-hard. Is it FPT? **Yes**

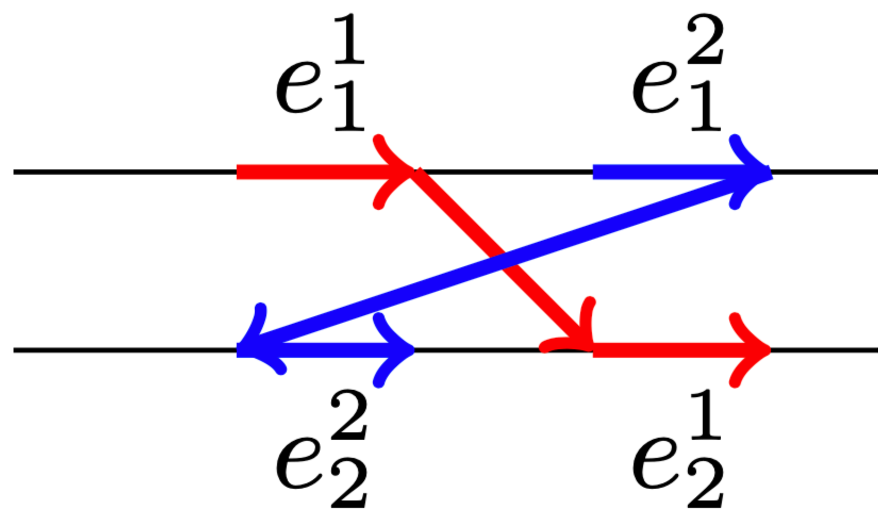


Boolean MinCSP and Cuts

3-Chain Sat

3-Chain SAT is FPT if the solution is a **minimum** st-cut.





Uncrossing

Boolean MinCSP and Cuts

3-Chain Sat

3-Chain SAT is FPT if the solution is a **minimum** st-cut.

What if it's **not a minimum** cut?

Boolean MinCSP and Cuts

Flow Augmentation Theorem (simple version)

There exists a polynomial-time algorithm that, given

- a directed graph G with $s, t \in V(G)$ and an integer k,

returns

- a set $A \subseteq V(G) \times V(G)$

such that for every minimal st-cut Z of size at most k, with probability $2^{-\mathcal{O}(k^4 \log k)}$

- Z is an st-cut of minimum cardinality in $G + A$.

From Eun-Jung Kim's slides for the 2023 Workshop on Logic, Graphs and Algorithms

Bijunctive Boolean Relations

When does Flow Augmentation help?

Domain $D = \{0,1\}$ is called Boolean

Examples of unary and binary Boolean relations:

- $(x = 0)$, $(x = 1)$ and $(x \vee y)$, $(\neg x \vee \neg y)$, $(x \to y)$, $(x = y)$, $(x \neq y)$

A bijunctive Boolean relation is an AND of unary and binary relations, e.g.

- $(x_1 = x_2) \wedge (x_3 = x_4)$
- $(x_1 \to x_2) \wedge (x_2 \to x_3) \wedge (x_3 \to x_4)$
- $(x_1 \to x_2) \wedge (x_3 \to x_4) \wedge (\neg x_1 \vee \neg x_3)$

Bijunctive Boolean MinCSP

A bijunctive Boolean relation is an AND of unary and binary relations, e.g.

- $(x_1 = x_2) \wedge (x_3 = x_4)$
- $(x_1 \to x_2) \wedge (x_2 \to x_3) \wedge (x_3 \to x_4)$
- $(x_1 \to x_2) \wedge (x_3 \to x_4) \wedge (\neg x_1 \vee \neg x_3)$

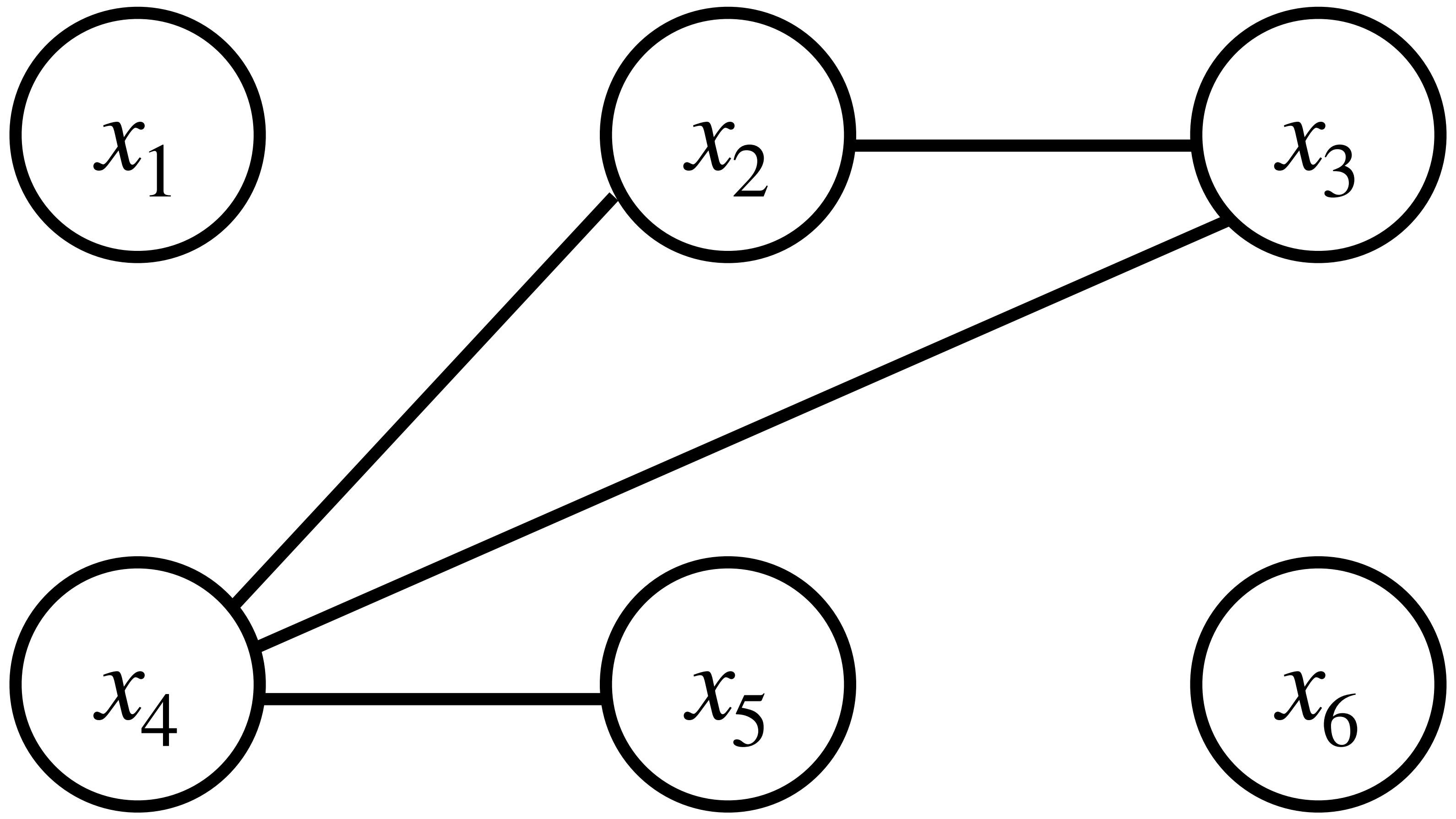
Theorem [Kim Kratsch Pilipczuk Wahlström 2023]:

If all relations in Γ are bijunctive and 2K2-free, then MinCSP(Γ) is in FPT.

2K2-free Bijunctive Relations

Formula: $(x_1 = 0) \wedge (x_2 \to x_3) \wedge (x_3 = x_4) \wedge (\neg x_2 \vee \neg x_4) \wedge (x_4 \neq x_5) \wedge (x_6 = 1)$

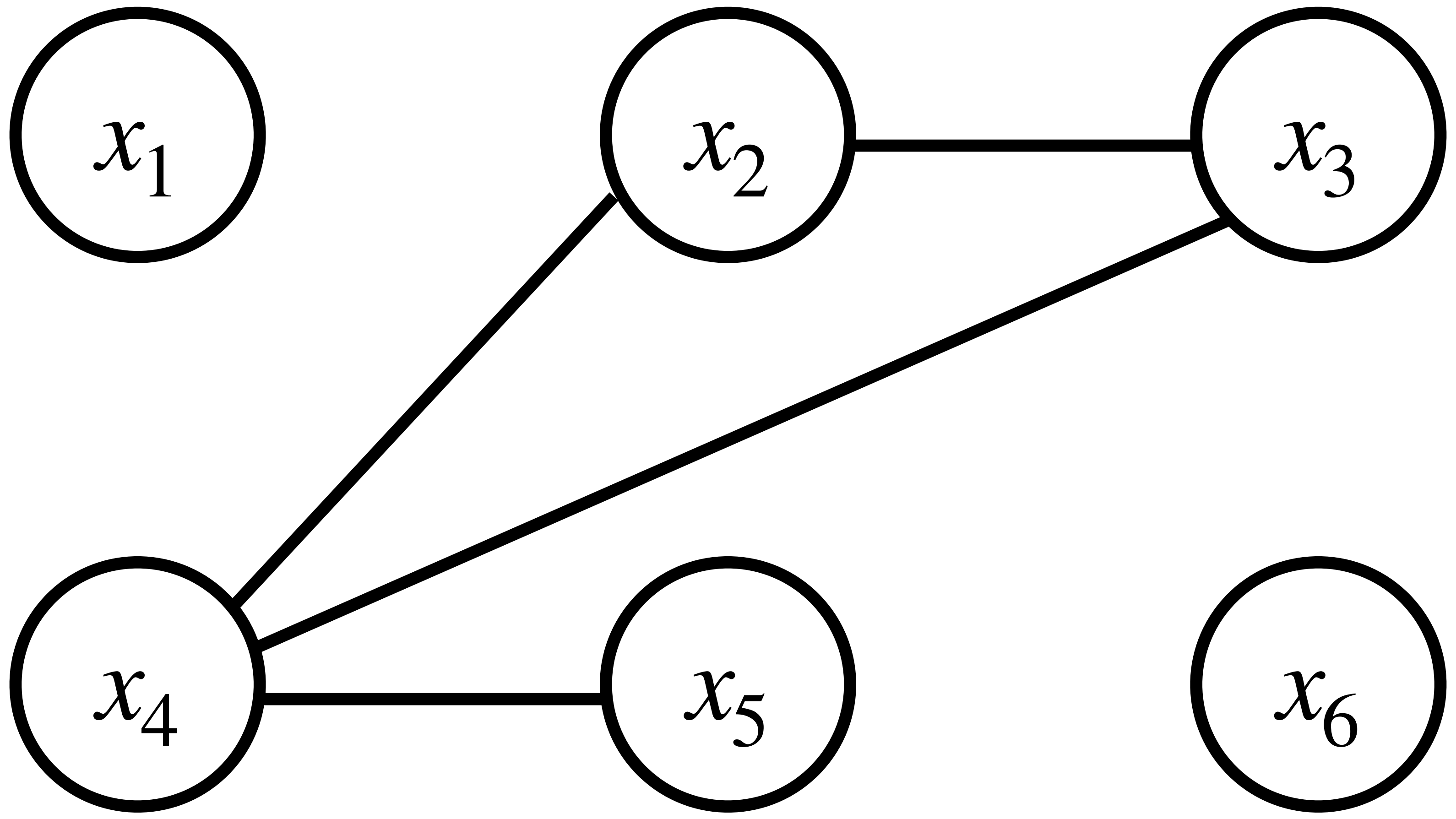
Gaifman graph:



2K2-free Bijunctive Relations

Formula: $(x_1 = 0) \wedge (x_2 \to x_3) \wedge (x_3 = x_4) \wedge (\neg x_2 \vee \neg x_4) \wedge (x_4 \neq x_5) \wedge (x_6 = 1)$

Gaifman graph: **2K2-free**



2K2-free Bijunctive Relations

Formula: $(x_1 \to x_2) \wedge (x_3 \neq x_4)$

Gaifman graph:

2K2-free Bijunctive Relations

Formula: $(x_1 \rightarrow x_2) \wedge (x_3 \neq x_4)$

Gaifman graph: **has a 2K2**

2K2-free Bijunctive Relations

A bijunctive relation R is 2K2-free if it is definable by a 2K2-free formula.

- $R(x_1, x_2, x_3, x_4) \equiv (x_1 \rightarrow x_2) \wedge (x_3 \rightarrow x_4)$ is **not** 2K2-free
- $R(x_1, x_2, x_3, x_4) \equiv (x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_3) \wedge (x_3 \rightarrow x_4)$ is 2K2-free
- $R(x_1, x_2, x_3, x_4, x_5) \equiv (x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_3) \wedge (x_3 \rightarrow x_4) \wedge (x_4 \rightarrow x_5)$ **?**

2K2-free Bijunctive Relations

A bijunctive relation R is 2K2-free if it is definable by a 2K2-free formula.

- $R(x_1, x_2, x_3, x_4) \equiv (x_1 \to x_2) \wedge (x_3 \to x_4)$ is **not** 2K2-free
- $R(x_1, x_2, x_3, x_4) \equiv (x_1 \to x_2) \wedge (x_2 \to x_3) \wedge (x_3 \to x_4)$ is 2K2-free
- $R(x_1, x_2, x_3, x_4, x_5) \equiv (x_1 \to x_2) \wedge (x_2 \to x_3) \wedge (x_3 \to x_4) \wedge (x_4 \to x_5)$ is 2K2-free

  $R(x_1, x_2, x_3, x_4, x_5) \equiv \bigwedge_{i<j}(x_i \to x_j)$ whose Gaifman graph is a 5-clique

Solving Point Algebra

Using Boolean MinCSP

Point Algebra

Time to deliver on the promise

Domain: $\mathbb{Q}$

Relations: subsets Γ of $<, \leq, =, \neq$

What is the complexity of MinCSP(Γ) for all such Γ?

- MinCSP($=, \neq$) is Edge Multicut **FPT**
- MinCSP($<$) is Directed Feedback Arc Set (DFAS) **FPT**
- MinCSP($<, \leq$) is Subset DFAS **FPT**

All FPT cases can be solved using Boolean MinCSP!

Point Algebra

Time to deliver on the promise

Domain: $\mathbb{Q}$

Relations: subsets Γ of $<, \leq, =, \neq$

What is the complexity of MinCSP(Γ) for all such Γ?

- **MinCSP($=, \neq$) is Edge Multicut** **FPT**
- MinCSP($<$) is Directed Feedback Arc Set (DFAS) **FPT**
- MinCSP($<, \leq$) is Subset DFAS **FPT**

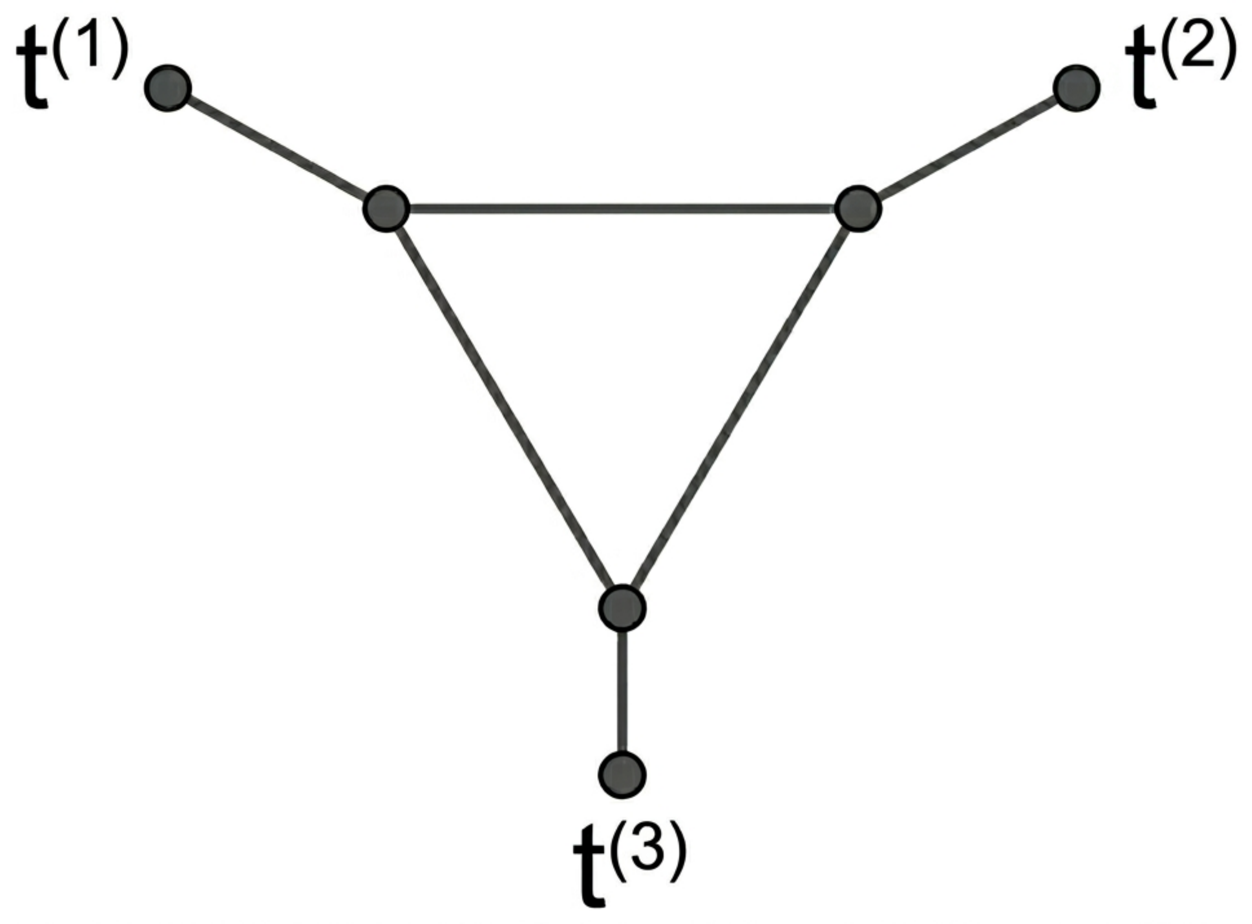
All FPT cases can be solved using Boolean MinCSP!

Solving Edge Multiway Cut

Warm-up

Input: a graph G, terminals $t^{(1)}, \ldots, t^{(\ell)} \in V(G)$, integer k.

Goal: find $Z \subseteq E(G)$ of size k such that each connected component of contains at most one terminal.



Solving Edge Multiway Cut

Input: a graph G, terminals $t^{(1)}, \ldots, t^{(\ell)} \in V(G)$, integer k.
Goal: find $Z \subseteq E(G)$ of size k such that each connected component of $G \backslash Z$ contains at most one terminal.

Boolean encoding:
For $v \in V(G)$, introduce Boolean variables $v_1, \ldots, v_\ell$

Semantics:
$v_i = 1$ if and only if the component of v in $G \backslash Z$ contains terminal $t^{(i)}$

Solving Edge Multiway Cut

Sketch

Input: a graph G, terminals $t^{(1)}, \ldots, t^{(\ell)} \in V(G)$, integer k.
Goal: find $Z \subseteq E(G)$ of size k such that each connected component of $G \backslash Z$ contains at most one terminal.

Boolean encoding:
For $v \in V(G)$, introduce Boolean variables $v_1, \ldots, v_\ell$

- $(t_i^{(i)} = 1)$ for all i
- $(\neg v_i \vee \neg v_j)$ for all $v \in V(G)$ and $i \neq j$
- $(u_1 = v_1) \wedge \ldots \wedge (u_\ell = v_\ell)$ for all $uv \in E(G)$

Solving Edge Multiway Cut

Sketch

Input: a graph G, terminals $t^{(1)}, \ldots, t^{(\ell)} \in V(G)$, integer k.
Goal: find $Z \subseteq E(G)$ of size k such that each connected component of $G \backslash Z$ contains at most one terminal.

Boolean encoding:
For $v \in V(G)$, introduce Boolean variables $v_1, \ldots, v_\ell$

- $(t_i^{(i)} = 1)$ for all i — **crisp** (undeletable, $k + 1$ copies)
- $(\neg v_i \vee \neg v_j)$ for all $v \in V(G)$ and $i \neq j$ — **crisp** (undeletable, $k + 1$ copies)
- $(u_1 = v_1) \wedge \ldots \wedge (u_\ell = v_\ell)$ for all $uv \in E(G)$ — **soft** (deletable at cost 1)

Solving Edge Multiway Cut

Sketch

Input: a graph G, terminals $t^{(1)}, \ldots, t^{(\ell)} \in V(G)$, integer k.
Goal: find $Z \subseteq E(G)$ of size k such that each connected component of $G \backslash Z$ contains at most one terminal.

Boolean encoding:

For $v \in V(G)$, introduce Boolean variables $v_1, \ldots, v_\ell$

- $(t_i^{(i)} = 1)$ for all i — **crisp** (undeletable, $k+1$ copies)
- $(\neg v_i \vee \neg v_j)$ for all $v \in V(G)$ and $i \neq j$ — **crisp** (undeletable, $k+1$ copies)
- $(u_1 = v_1) \wedge \ldots \wedge (u_\ell = v_\ell)$ for all $uv \in E(G)$ — **soft** (deletable at cost 1)

Not 2K2-free!

Solving Edge Multiway Cut

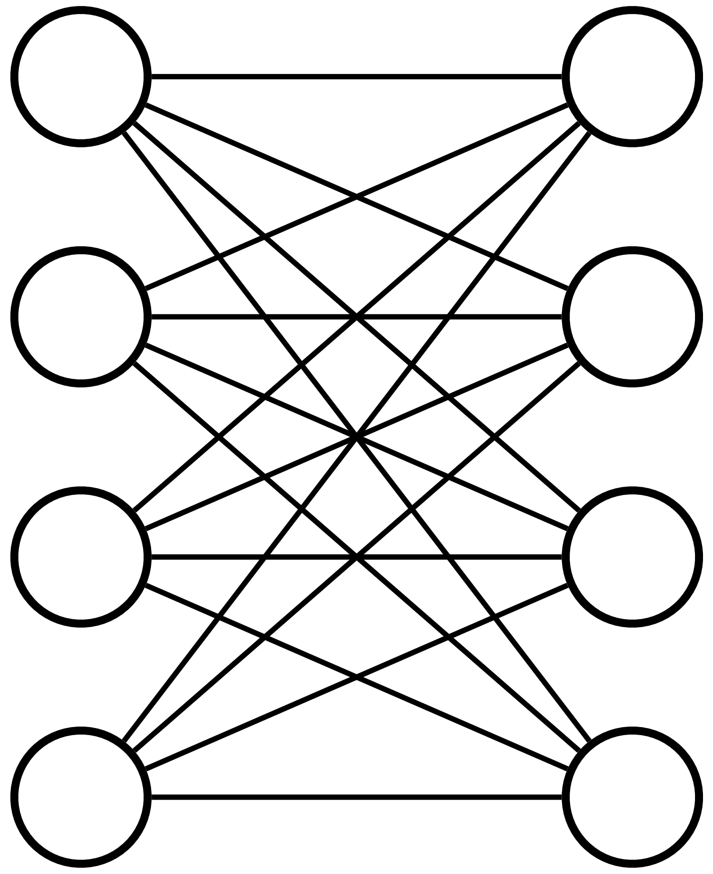
Full Encoding

Input: a graph G, terminals $t^{(1)}, \ldots, t^{(\ell)} \in V(G)$, integer k.
Goal: find $Z \subseteq E(G)$ of size k such that each connected component of $G \backslash Z$ contains at most one terminal.

Fix: for every edge $uv \in E(G)$, introduce a soft constraint:

$$\bigwedge_{i=1}^{\ell} (u_i = v_i) \wedge \bigwedge_{i \neq j} (\neg u_i \vee \neg v_j)$$

2K2-free!



Solving Edge Multiway Cut

Sketch

Input: a graph G, terminals $t^{(1)}, \ldots, t^{(\ell)} \in V(G)$, integer k.
Goal: find $Z \subseteq E(G)$ of size k such that each connected component of $G \backslash Z$ contains at most one terminal.

Boolean encoding:
For $v \in V(G)$, introduce Boolean variables $v_1, \ldots, v_\ell$

- $(t_i^{(i)} = 1)$ for all i **crisp**
- $(\neg v_i \vee \neg v_j)$ for all $v \in V(G)$ and $i \neq j$ **crisp**
- $\bigwedge_{i=1}^{\ell}(u_i = v_i) \wedge \bigwedge_{i \neq j}(\neg u_i \vee \neg v_j)$ for all $uv \in E(G)$ **soft**

Solving Edge Multiway Cut ++

Input: a graph G, terminals $T \subseteq V(G)$, a set $\mathscr{P}$ of vertex pairs.

Goal: find $Z \subseteq E(G)$ of size k such that

(1) each connected component of $G \backslash Z$ contains at most one terminal, and

(2) each connected component of $G \backslash Z$ with a terminal contains at most one vertex from every pair $\{p, q\} \in \mathscr{P}$.

Solving Edge Multicut

Input: a graph G, terminals $T \subseteq V(G)$, a set $\mathscr{P}$ of vertex pairs.

Goal: find $Z \subseteq E(G)$ of size k such that
(1) each connected component of $G \backslash Z$ contains at most one terminal, and
(2) each connected component of $G \backslash Z$ with a terminal contains at most one vertex from every pair $\{p, q\} \in \mathscr{P}$.

$$\text{Edge Multicut} \xrightarrow{\textit{iterative compression, FPT guessing}} \text{Edge Multiway Cut ++}$$

Solving Edge Multicut

Input: a graph G, a set $\mathscr{P}$ of vertex pairs.

Goal: find $Z \subseteq E(G)$ of size k such that
$G \backslash Z$ contains no st-path for any $\{s, t\} \in \mathscr{P}$

Edge Multicut $\xrightarrow{\textit{iterative compression, FPT guessing}}$ Edge Multiway Cut ++

Solving Edge Multiway Cut ++

Input: a graph G, terminals $T \subseteq V(G)$, a set $\mathscr{P}$ of vertex pairs.

Goal: find $Z \subseteq E(G)$ of size k such that
(1) each connected component of $G \backslash Z$ contains at most one terminal, and
(2) each connected component of $G \backslash Z$ with a terminal contains at most one vertex from every pair $\{p, q\} \in \mathscr{P}$.

Use the Edge Multiway Cut encoding and add a crisp constraint

$(\neg p_i \vee \neg q_i)$ for every pair $\{p, q\} \in \mathscr{P}$ and every i.

One-hot Encoding

Variable x, possible states $\{1, \ldots, d\}$

Boolean encoding: $x_1, \ldots, x_d$ with possible states $(0,0,\ldots,0)$, $(1,0,\ldots,0)$, $(0,1,\ldots,0)$, …, $(0,0,\ldots,1)$

How to enforce? $(\neg x_i \vee \neg x_j)$ for all $i \neq j$

- $(0,0,\ldots,0)$: x takes a default value (usually from iterative compression)
- $(1,0,\ldots,0)$: $x = 1$
- $(0,1,\ldots,0)$: $x = 2$
- $\ldots$
- $(0,0,\ldots,1)$: $x = d$

One-hot Encoding

Applications:

- Edge Multicut
- Vertex Multicut [OW ESA'24]
- Triple Multicut [OW ESA'24]
- Used in the Min-2-Lin($\mathbb{Z}$) algorithm [DJOOW SODA'23]
- Used in the MinCSP($x = y + 1,\ x \neq y$) algorithm [DJOOdV KR'26]

Point Algebra

Domain: $\mathbb{Q}$

Relations: subsets Γ of $<, \leq, =, \neq$

What is the complexity of MinCSP(Γ) for all such Γ?

- MinCSP($=, \neq$) is Edge Multicut **FPT**
- **MinCSP($<$) is Directed Feedback Arc Set (DFAS) FPT**
- MinCSP($<, \leq$) is Subset DFAS **FPT**

All FPT cases can be solved using Boolean MinCSP!

Solving Directed Feedback Arc Set (DFAS)

Input: a directed graph G, integer k.

Goal: find $Z \subseteq A(G)$ of size k such that $G \backslash Z$ is acyclic.

Solving DFAS

After Iterative Compression and Guessing

Input: a directed graph G, integer k **and**
$\{x_1, \ldots, x_{k+1}\} = X \subseteq V(G)$ such that $G - X$ is acyclic.

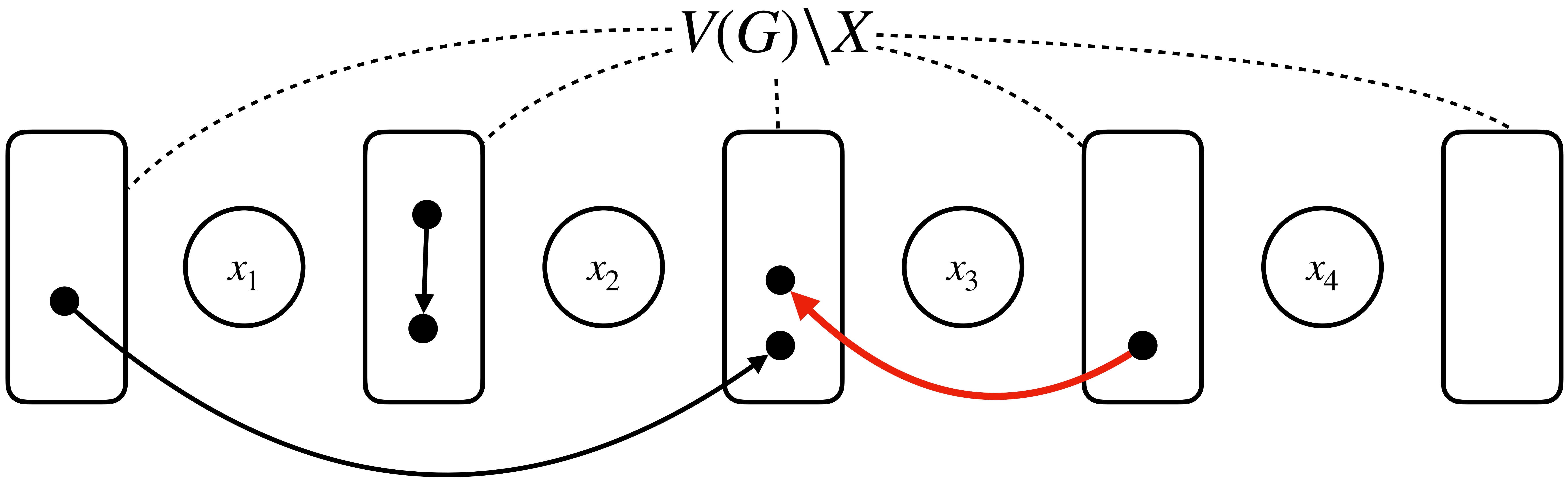
Goal: find $Z \subseteq A(G)$ of size k such that $G \backslash Z$ is acyclic **and**
$G \backslash Z$ admits a topological ordering $\prec$ such that $x_1 \prec x_2 \prec \ldots \prec x_{k+1}$.

Solving DFAS

After Iterative Compression and Guessing

Input: a directed graph G, integer k **and**
$\{x_1, \ldots, x_{k+1}\} = X \subseteq V(G)$ such that $G - X$ is acyclic.

Goal: find $Z \subseteq A(G)$ of size k such that $G \backslash Z$ is acyclic **and**
$G \backslash Z$ admits a topological ordering $\prec$ such that $x_1 \prec x_2 \prec \ldots \prec x_{k+1}$.

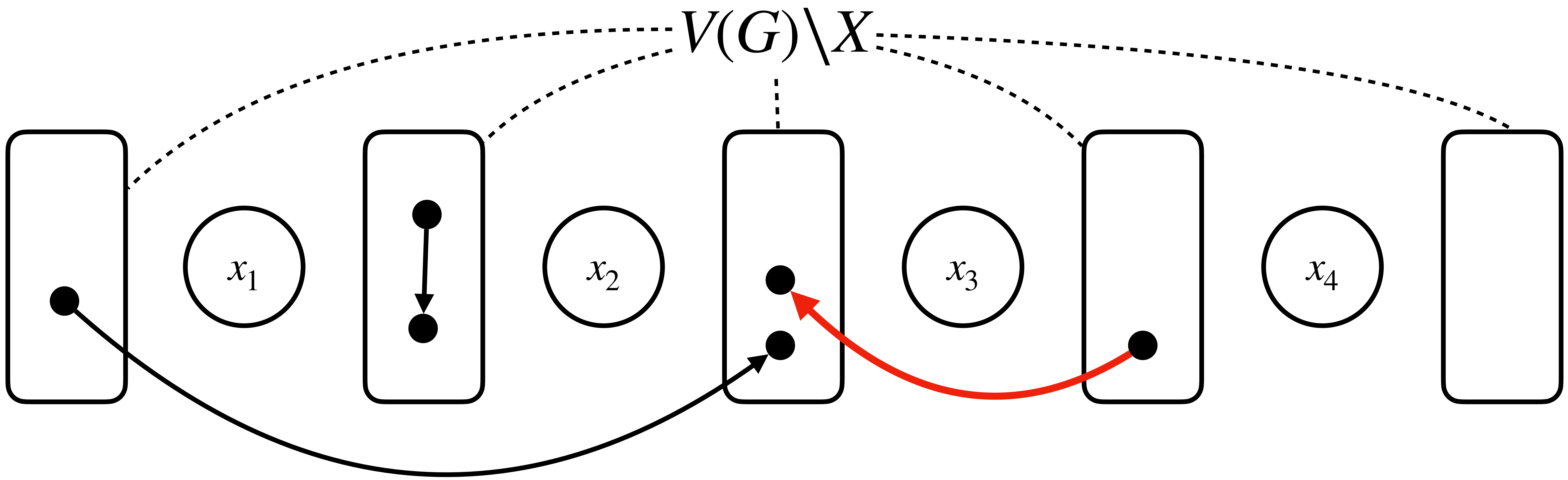


Solving DFAS

After Iterative Compression and Guessing

For every $v \in V(G) \backslash X$, create Boolean variables $v[1], \ldots, v[k+1]$

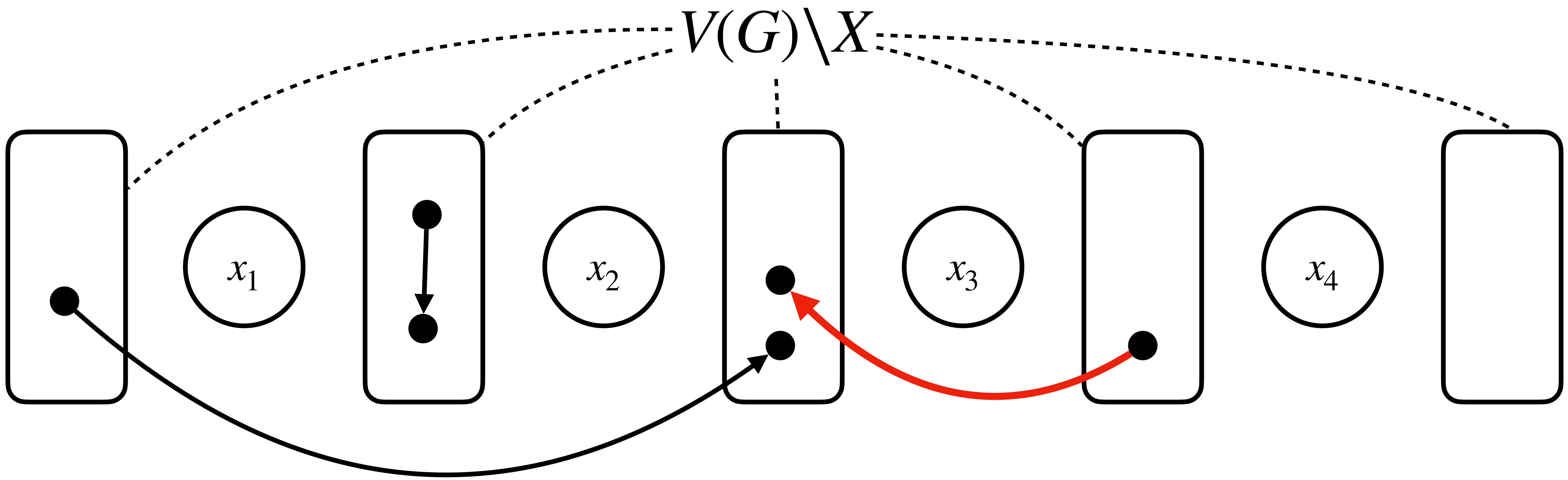
Semantics: $v[i] = 1$ if and only if $v \leq x_i$



Solving DFAS

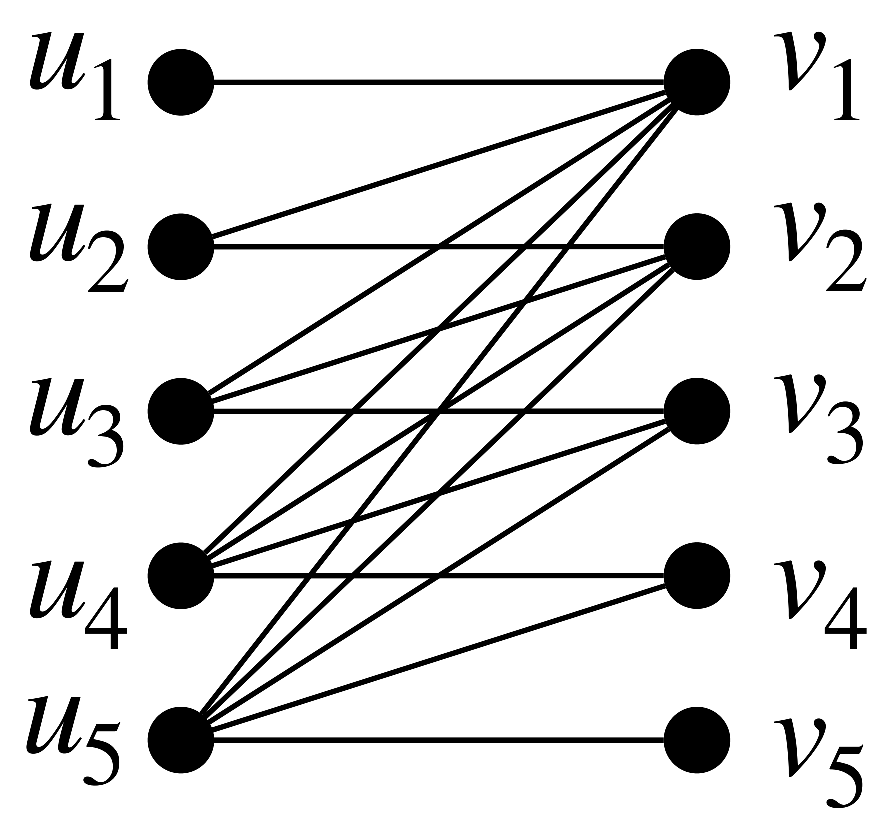
After Iterative Compression and Guessing

- For all $i < j$, add $(v[i] \leftarrow v[j])$ // possible states $(0{,}0{,}\ldots{,}0)$, $(1{,}0{,}\ldots{,}0)$, $(1{,}1{,}\ldots{,}0)$, …, $(1{,}1{,}\ldots{,}1)$
- For all x_i, add $(x_i[j] = 1)$ for $j \leq i$, and $(x_i[j] = 0)$ for $j > i$. // state of x_2 is $(1{,}1{,}0{,}\ldots{,}0)$



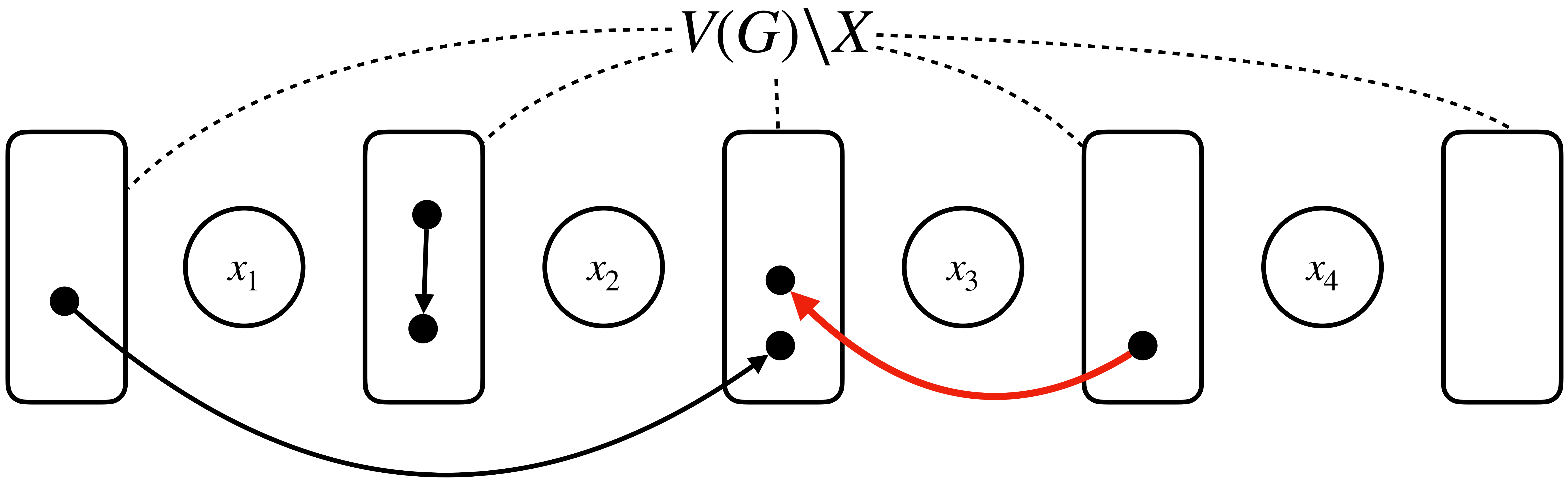
Solving DFAS

After Iterative Compression and Guessing



Soft constraints:

For every arc $(u, v) \in A(G)$, add $\bigwedge_{i \leq j} (u[i] \rightarrow v[j])$.



Unary Encoding

Variable x, possible states $\{1, \ldots, d\}$

Boolean encoding: $x_1, \ldots, x_d$ with possible states $(0,0,0,\ldots,0)$, $(1,0,0,\ldots,0)$, $(1,1,0,\ldots,0)$, …, $(1,1,1,\ldots,1)$

How to enforce? $(x_i \rightarrow x_j)$ for all $i \leq j$

- $(0,0,\ldots,0)$: x takes a default value (usually from iterative compression)
- $(1,0,\ldots,0)$: $x = 1$
- $(1,1,\ldots,0)$: $x = 2$
- $\ldots$
- $(1,1,\ldots,1)$: $x = d$

Unary Encoding

Applications:

- DFAS
- Subset DFAS [KMPSW SIDMA'24] (multi-budget versions too)
- Tractable fragments of Interval Algebra [DJOOPS IPEC'23]
- List H-Coloring by Vertex Deletion [CEM ESA'13]

One-Hot vs Unary Encoding

One-hot encoding supports $=$ and $\neq$.

Unary encoding supports $\leq$ and $<$.

Combine? MinCSP($=$, $\neq$, $<$) is **FPT**.

But MinCSP($\leq$, $\neq$) is **W[1]-hard**.

Is Boolean encodings all we need?

No.

- Directed Multicut with 3 Terminal Pairs (tune in for Roohani's talk)
- Min-2-Lin($\mathbb{Z}_4$) [DJOOW FOCS'26]

Open Problems

Open Problem 1

Edge Deletion List Homomorphism

LHomED(H)

Input: G, $L : V(G) \to 2^{V(H)}$, k.

Goal: remove k edges from G so that it becomes L-homomorphic to H.

Open Problem 1

Edge Deletion List Homomorphism

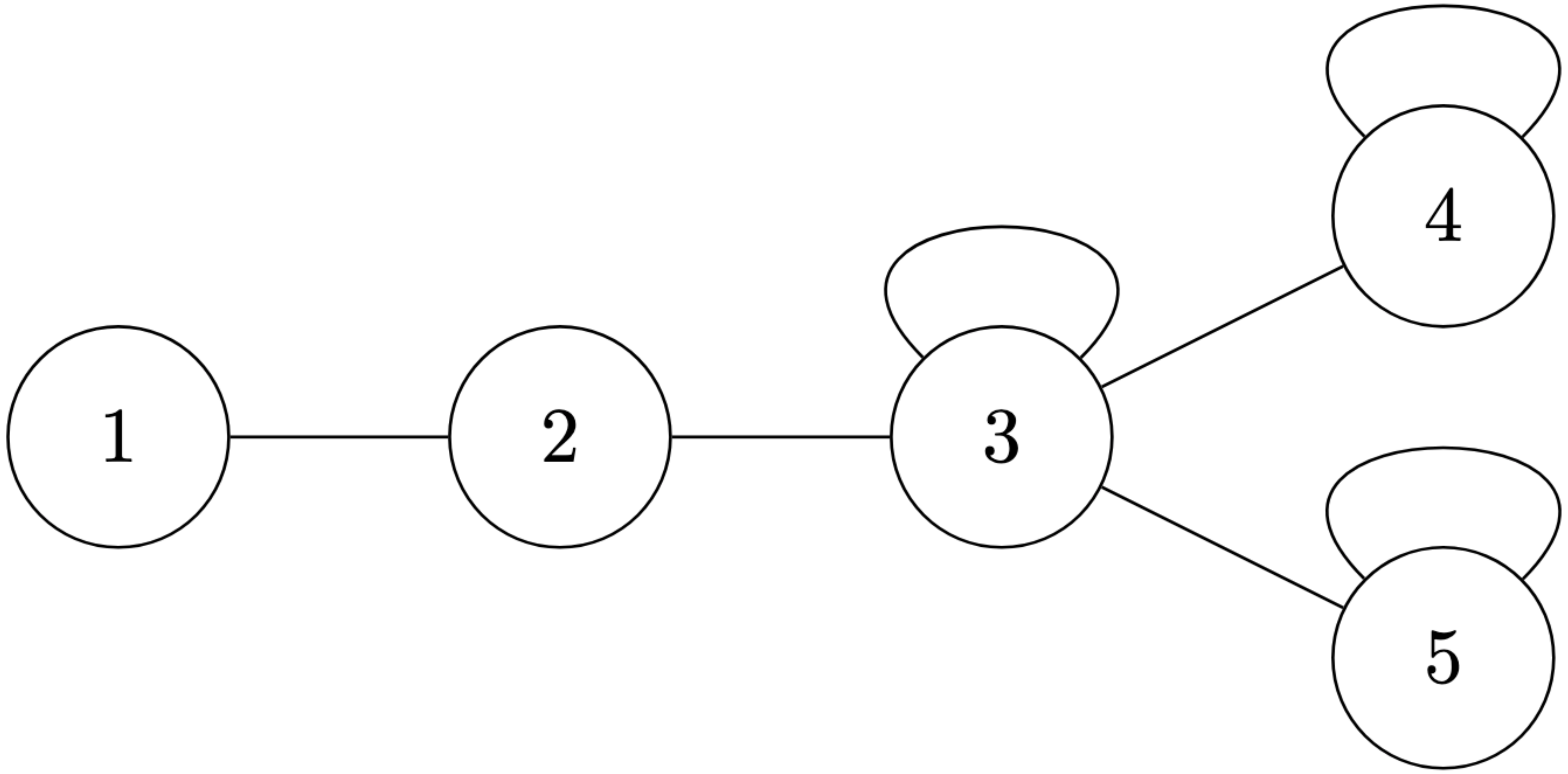
LHomED(H)

Input: G, $L : V(G) \to 2^{V(H)}$, k.

Goal: find $h : V(G) \to V(H)$ that preserves all but $\leq k$ edges and satisfies $h(v) \in L(v)$ for all $v \in V(G)$.

Open Problem 1

Edge Deletion List Homomorphism for the Dragon Graph

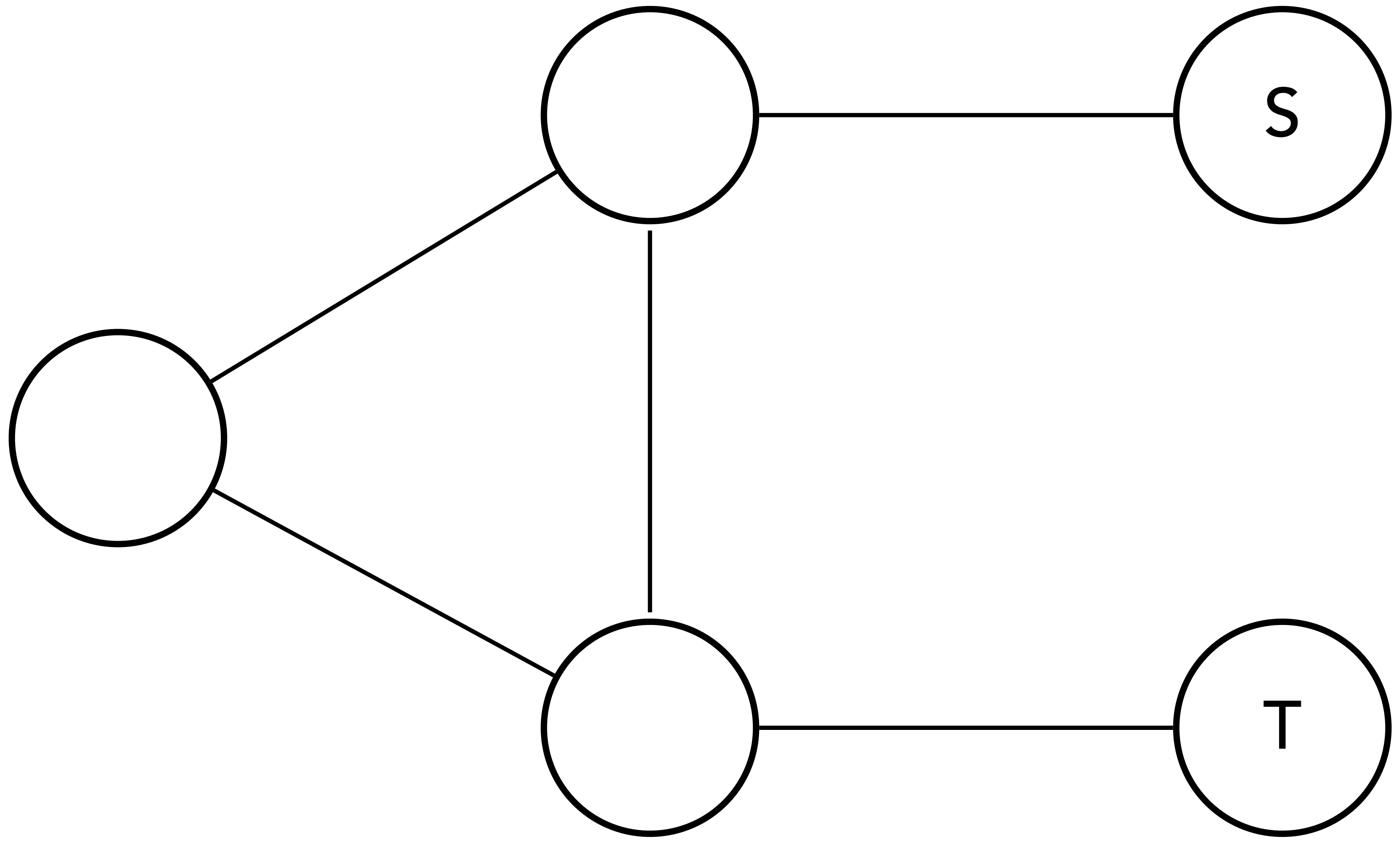


Open Problem 2

d-Hypergraph Strong st-Separator

Input: a d-uniform hypergraph H, vertices $s, t \in V(H)$, integer k.

Goal: remove k vertices **together with all incident hyperedges** so that there remains no st-path in H.



In P for $d = 2$.

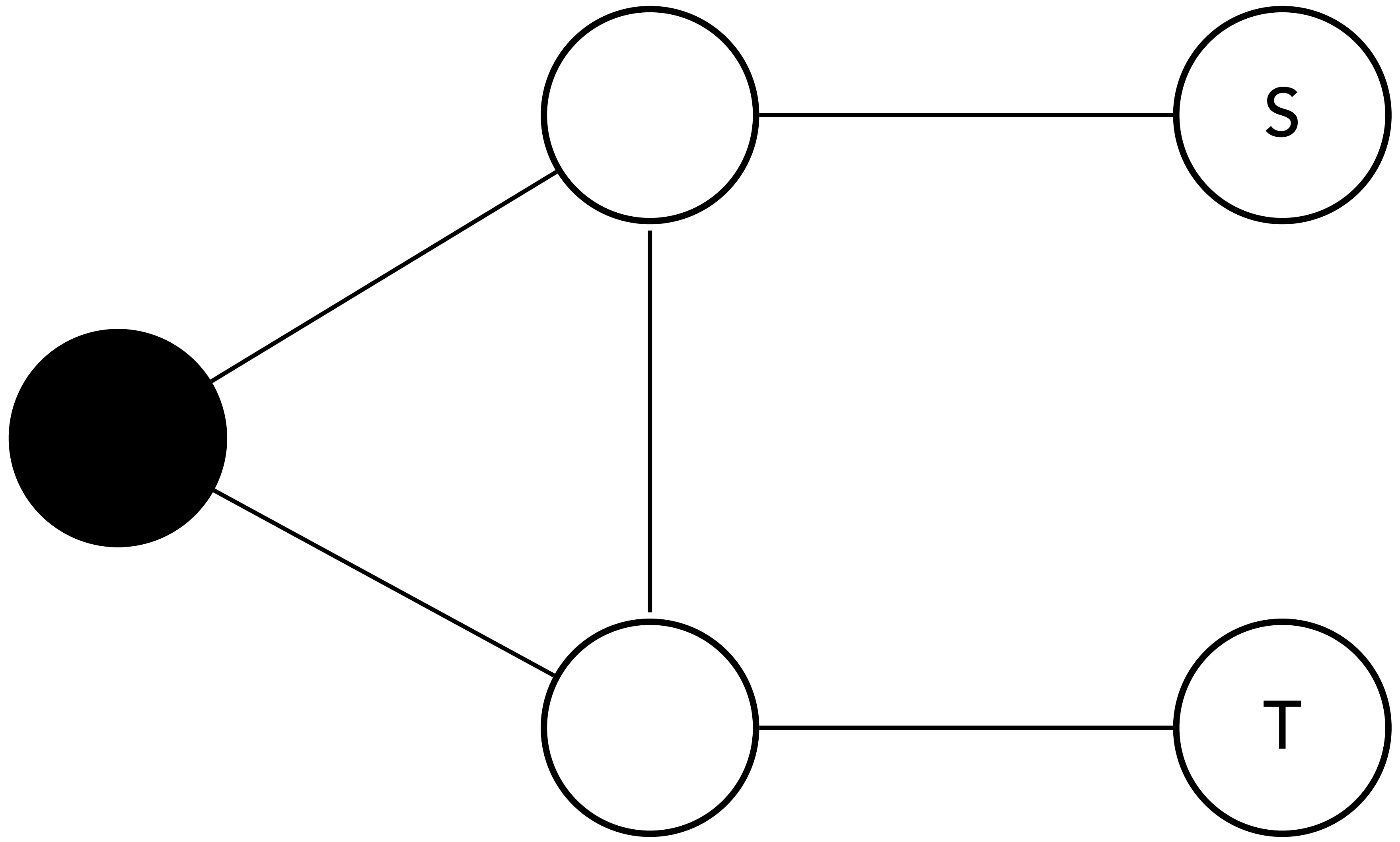
NP-hard for $d \geq 3$. FPT?

Open Problem 2

d-Hypergraph Strong st-Separator

Input: a d-uniform hypergraph H, vertices $s, t \in V(H)$, integer k.

Goal: remove k vertices **together with all incident hyperedges** so that there remains no st-path in H.



In P for $d = 2$.

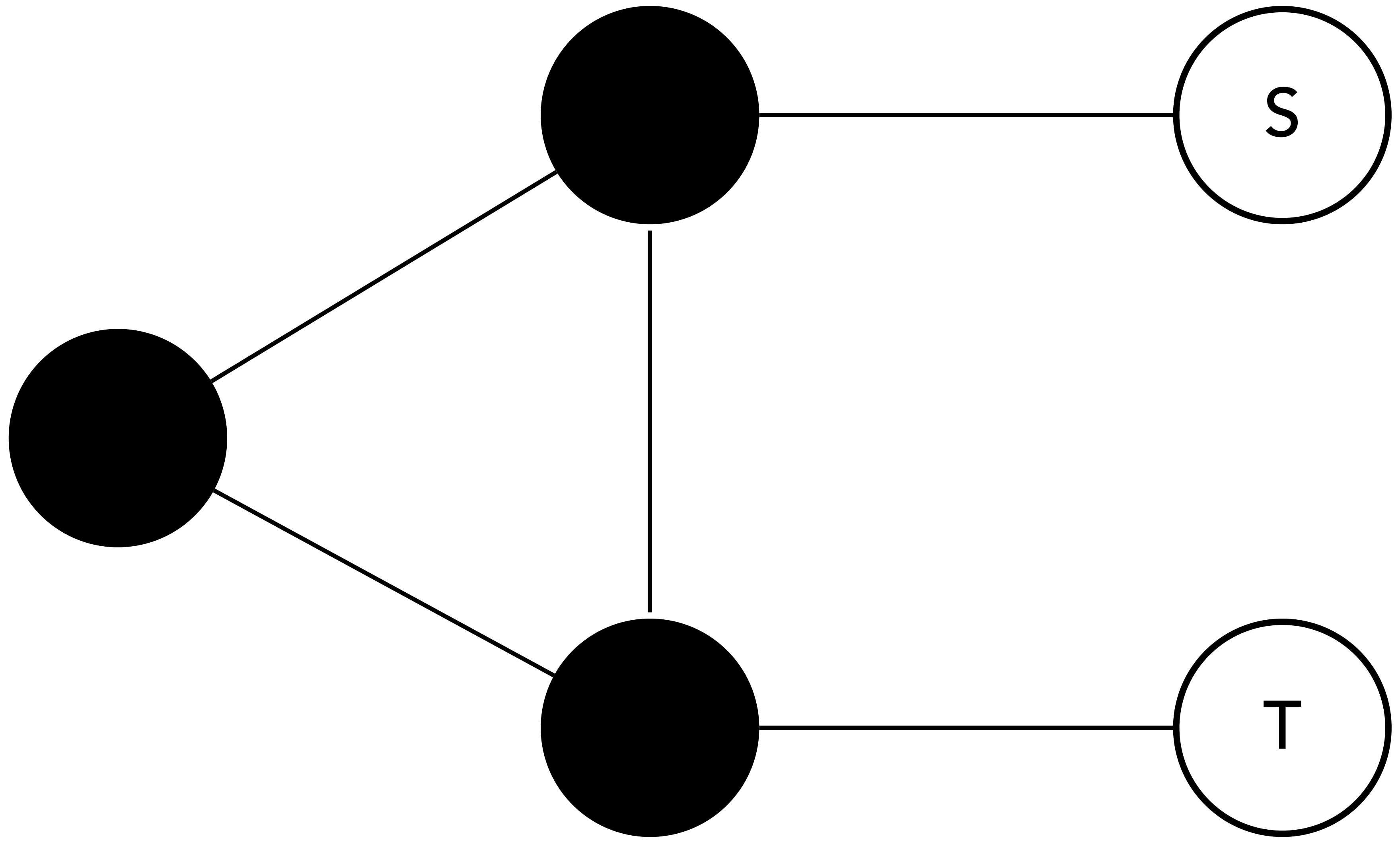
NP-hard for $d \geq 3$. FPT?

Open Problem 2

d-Hypergraph Strong st-Separator

Input: a d-uniform hypergraph H, vertices $s, t \in V(H)$, integer k.

Goal: remove k vertices **together with all incident hyperedges** so that there remains no st-path in H.



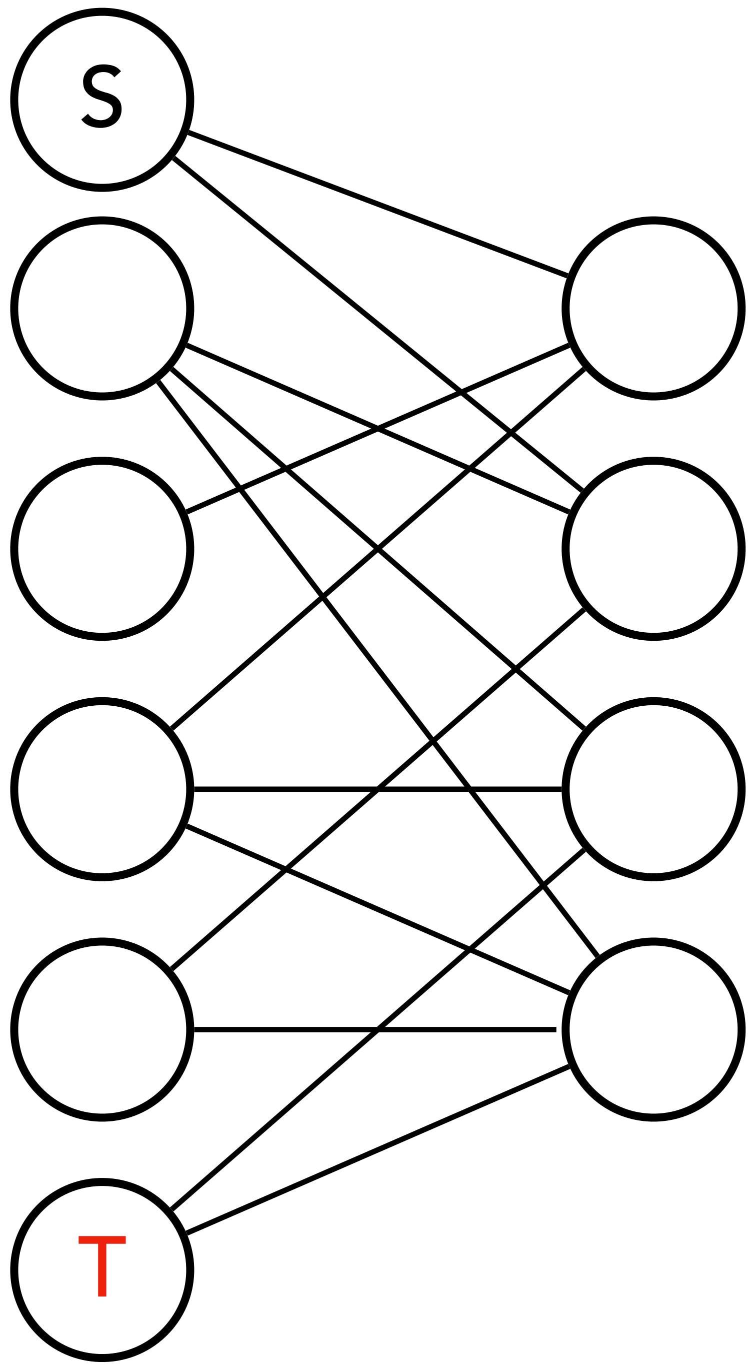
In P for $d = 2$.

NP-hard for $d \geq 3$. FPT?

Open Problem 2

d-Hypergraph Strong st-Separator

Choose k left vertices

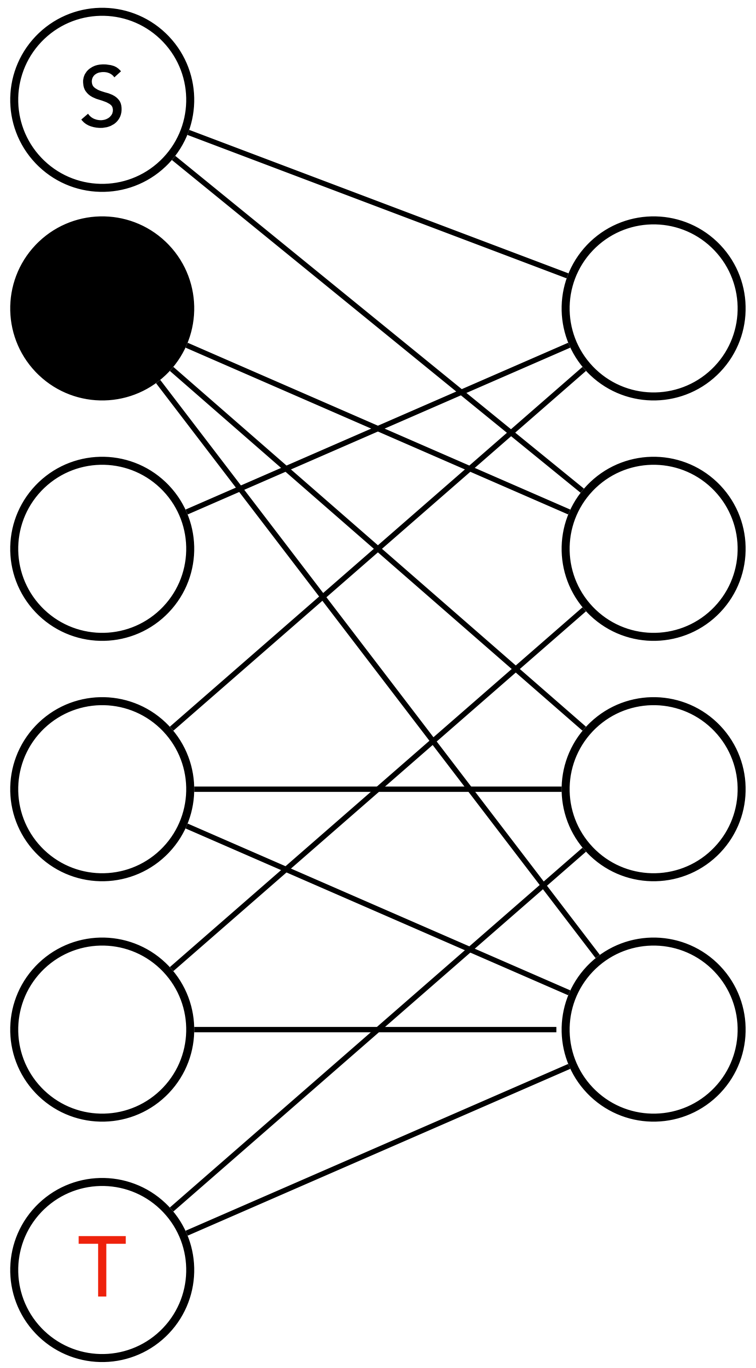


Right degree at most 3

Open Problem 2

d-Hypergraph Strong st-Separator

Choose k left vertices

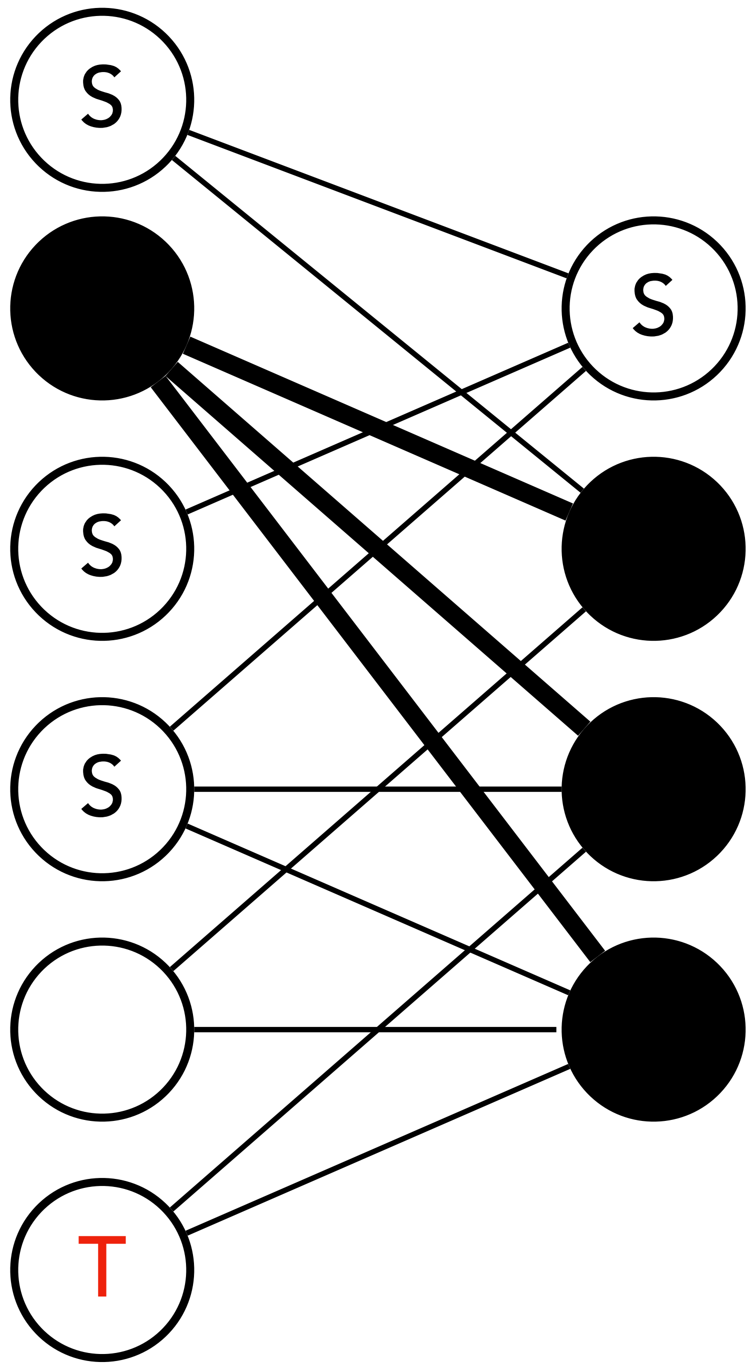


Right degree at most 3

Open Problem 2

d-Hypergraph Strong st-Separator

Choose k left vertices so that the closed neighborhood forms an st-cut



Right degree at most 3

Open Problem 3

Approximating Negative DFAS

Input: a digraph G, a labelling $\lambda : A(G) \to \{-1,0,1\}$, integer k.

Goal: remove k edges from G so that label sum on every cycle is ≥ 0.

Open Problem 3

Approximating Negative DFAS

Input: a digraph G, a labelling $\lambda : A(G) \to \{-1,0,1\}$, integer k.

Goal: remove k edges from G so that label sum on every cycle is ≥ 0.

$\lambda : A(G) \to \{-1\}$ is DFAS

$\lambda : A(G) \to \{-1,0\}$ is Subset DFAS

$\lambda : A(G) \to \{-1,0,1\}$ is W[1]-hard (Bérczi, Göke, Mendoza-Cadena, Mnich)

Open Problem 3

Approximating Negative DFAS

Input: a digraph G, a labelling $\lambda : A(G) \to \{-1,0,1\}$, integer k.

Goal: remove k edges from G so that label sum on every cycle is ≥ 0.

Can we FPT-approximate it within any constant?

Open Problems

- LHomED(Dragon Graph)
- d-Hypergraph Strong st-Separator
- Approximating Negative DFAS
- More at: https://pacs2024.github.io/pacs2024-open-problems.pdf

THANK YOU!