Colored Constrained Spanning Tree on Directed Graphs

Hung-Yeh Lee, Hsuan-Yu Liao and Wing-Kai Hon

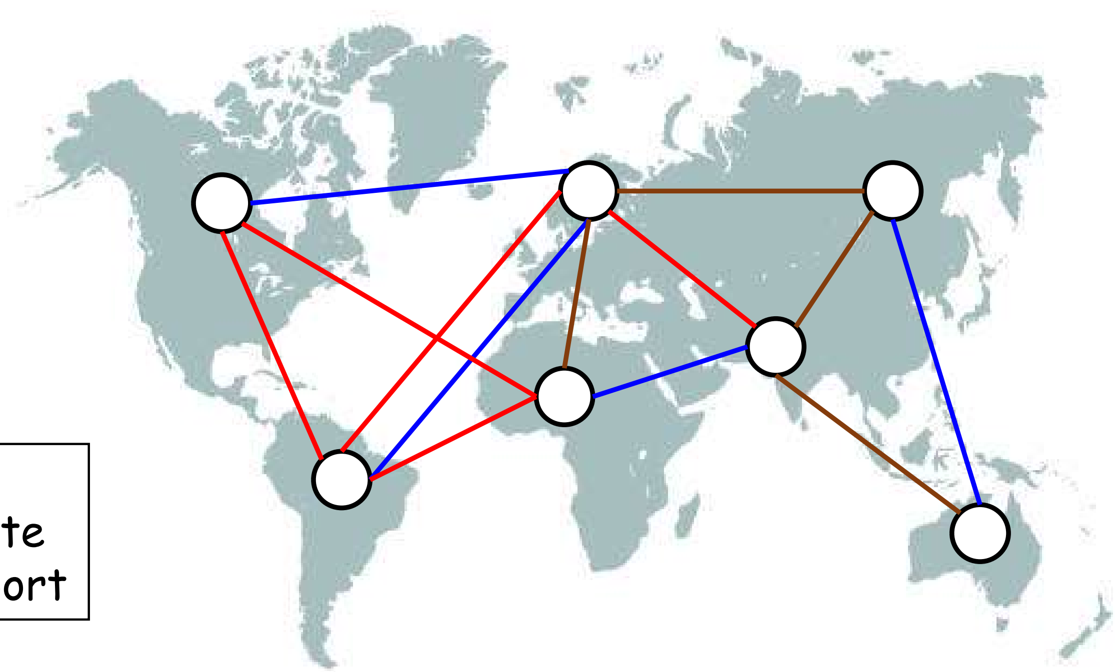
國立清華大學
NATIONAL TSING HUA UNIVERSITY

Introduction

Airline A

Airline B

Airline C



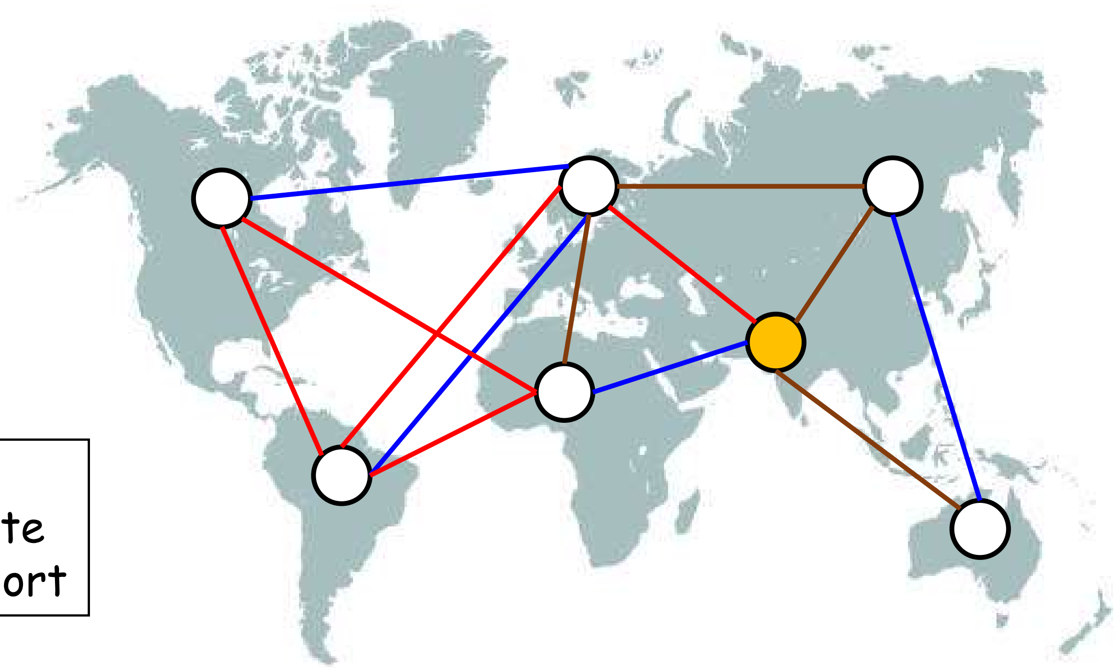
Constraint:
Each airline can only operate at most κ flights in an airport

Introduction

Airline A

Airline B

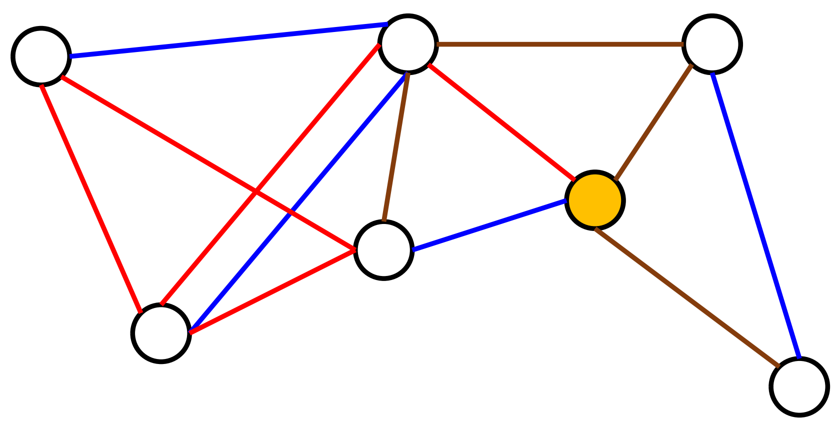
Airline C



Constraint:
Each airline can only operate
at most κ flights in an airport

κ-Colored-Constrained Spanning Tree (κ-CCST)

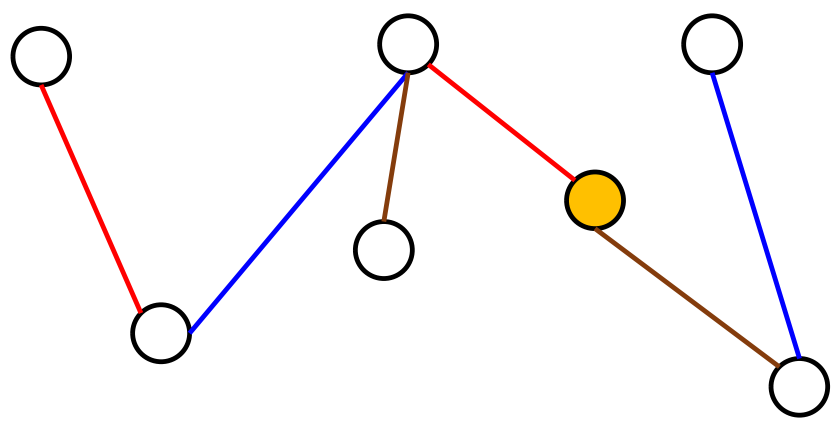
- κ-Colored-Constraint:
  the number of incident edges of the same color $\leq \kappa$



Example: 1-CCST

κ-Colored-Constrained Spanning Tree (κ-CCST)

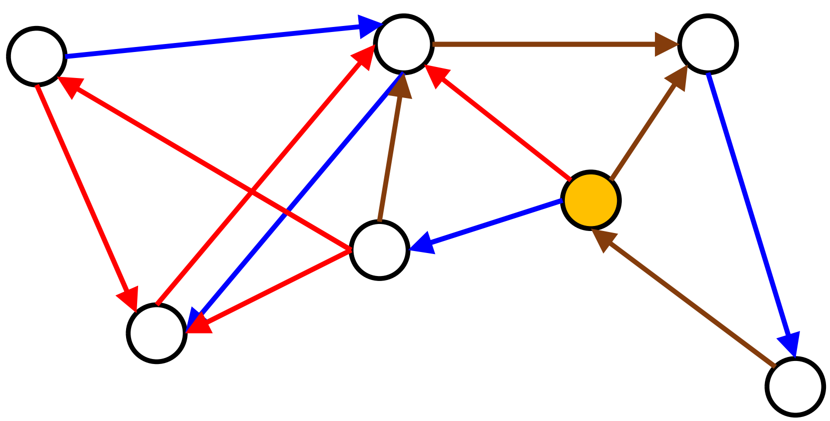
- κ-Colored-Constraint:
  the number of incident edges of the same color $\leq \kappa$



Example: 1-CCST

κ-Colored-Constrained Spanning Tree (κ-CCST)

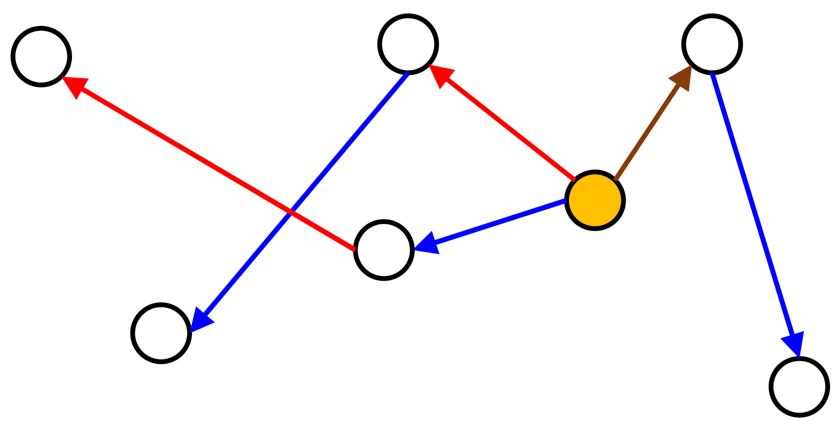
- Directed κ-CCST:
  A spanning out-tree that follows the κ-colored-constraint
- Both incoming and outgoing edges are counted



Example: directed 1-CCST

κ-Colored-Constrained Spanning Tree (κ-CCST)

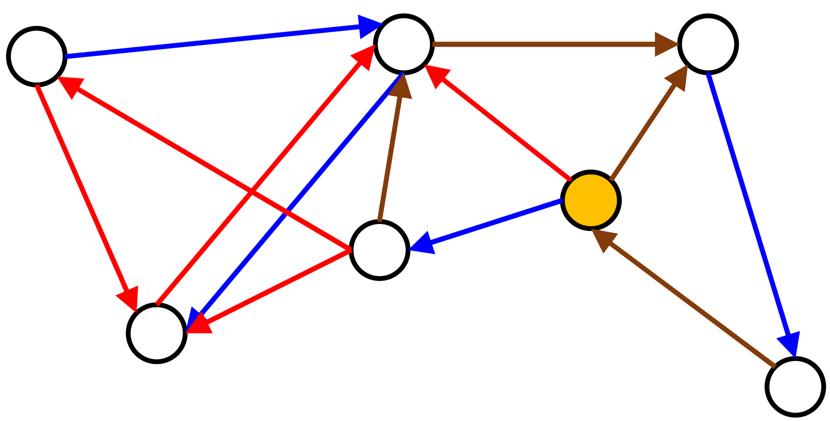
- Directed κ-CCST:
  A spanning out-tree that follows the κ-colored-constraint
- Both incoming and outgoing edges are counted



Example: directed 1-CCST

κ-Colored-Out-Constrained Spanning Tree (κ-COCST)

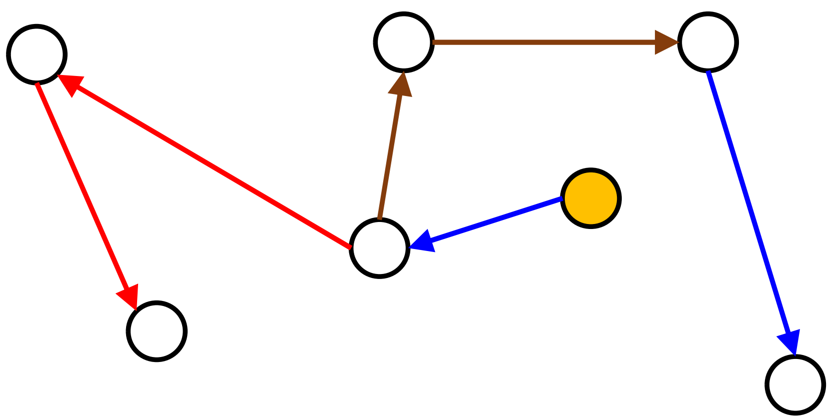
- κ-Colored-Out-Constraint:
  the number of outgoing edges of the same color $\leq \kappa$



Example: 1-COCST

κ-Colored-Out-Constrained Spanning Tree (κ-COCST)

- κ-Colored-Out-Constraint:
  the number of outgoing edges of the same color $\leq \kappa$



Example: 1-COCST

Previous Works

- 1-CCST on edge-colored undirected graphs [Borozan et al. (Eur. J. Comb. 19)]
    - NP-hard
    - Tractable in special cases
- Ramsey-type problems: [Kano et al. (Discrete Math. 20) …]
    - How large should the graph be to guarantee a 1-CCST

Results

- κ-CCST

Graph Class	λ	κ	Hardness
Directed	≥ 2	≥ 1	NP-Hard
DAG	≥ 3	≥ 1	NP-Hard
DAG	$= 2$	$= 1$	P

- κ-COCST

Graph Class	λ	κ	Hardness
Directed	≥ 4	≥ 1	NP-Hard
DAG	≥ 2	≥ 1	P

Results

- κ-CCST

Graph Class	λ	κ	Hardness
Directed	≥ 2	≥ 1	NP-Hard
DAG	≥ 3	≥ 1	NP-Hard
DAG	$= 2$	$= 1$	P

- κ-COCST

Graph Class	λ	κ	Hardness
Directed	≥ 4	≥ 1	NP-Hard
DAG	≥ 2	≥ 1	P

Results

- κ-CCST

Graph Class	λ	κ	Hardness
Directed	≥ 2	≥ 1	NP-Hard
DAG	≥ 3	≥ 1	NP-Hard
DAG	$= 2$	$= 1$	P

- κ-COCST

Graph Class	λ	κ	Hardness
Directed	≥ 4	≥ 1	NP-Hard
DAG	≥ 2	≥ 1	P

Results

- κ-CCST

Graph Class	λ	κ	Hardness
Directed	≥ 2	≥ 1	NP-Hard
DAG	≥ 3	≥ 1	NP-Hard
DAG	$= 2$	$= 1$	P

- κ-COCST

Graph Class	λ	κ	Hardness
Directed	≥ 4	≥ 1	NP-Hard
DAG	≥ 2	≥ 1	P

Results

- κ-CCST

Graph Class	λ	κ	Hardness
Directed	≥ 2	≥ 1	NP-Hard
DAG	≥ 3	≥ 1	NP-Hard
DAG	$= 2$	$= 1$	P

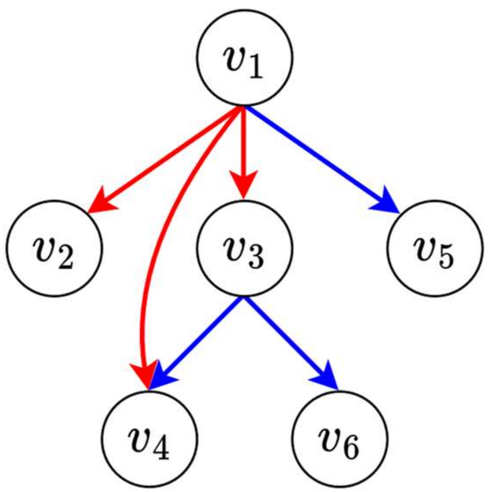
- κ-COCST

Graph Class	λ	κ	Hardness
Directed	≥ 4	≥ 1	NP-Hard
DAG	≥ 2	≥ 1	P

Tractable Cases

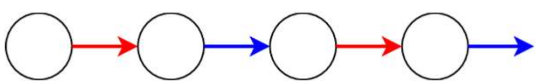
Rooted DAG

- Root: A vertex that can reach all vertices
- The root must be the only zero-indegree vertex on a DAG

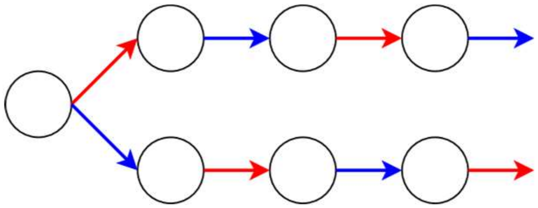


1-CCST on a 2-edge-colored DAGs

- Observe that a 1-CCST on a 2-edge-colored DAG must be either an *alternating path* or a *V-shaped tree*
- Dynamic Programming



alternating path

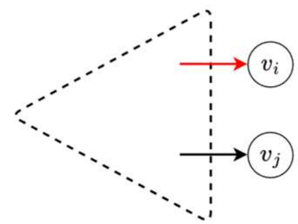


V-shaped Tree

The Dynamic Programming

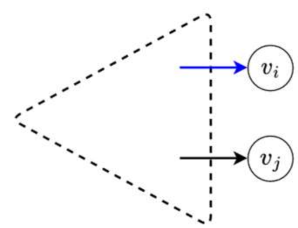
- Maintain the leaves and their incoming edge colors on trees spanning $v_1, v_2, \ldots, v_i$ for all $i = 1, 2, \ldots, n$

$f(v_i^{\text{red}}) = \{v_j^c : \text{tree}$



$\text{exists}\}$

$f(v_i^{\text{blue}}) = \{v_j^c : \text{tree}$

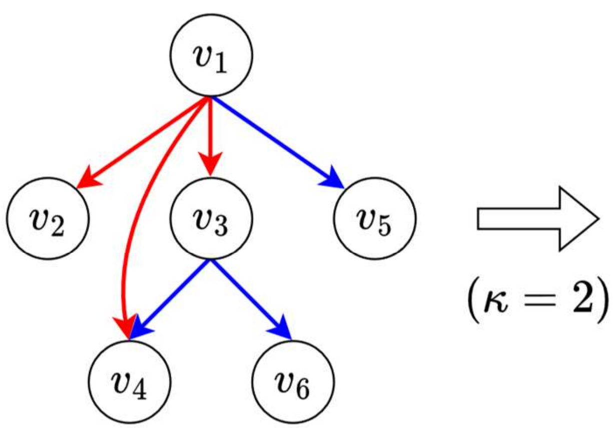


$\text{exists}\}$

- Allows a linear time algorithm using stack and bit vector

κ-COCST on λ-edge-colored DAGs

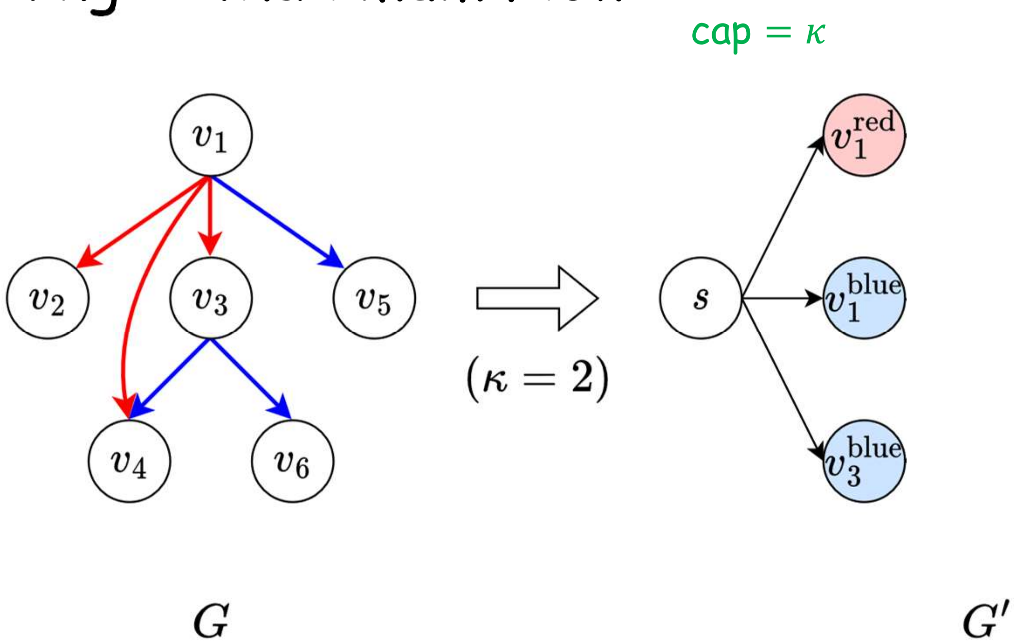
- Match each non-root vertex to one of its incoming edge, such that colored-out-constraint holds
- B-matching → Maximum Flow



G G'

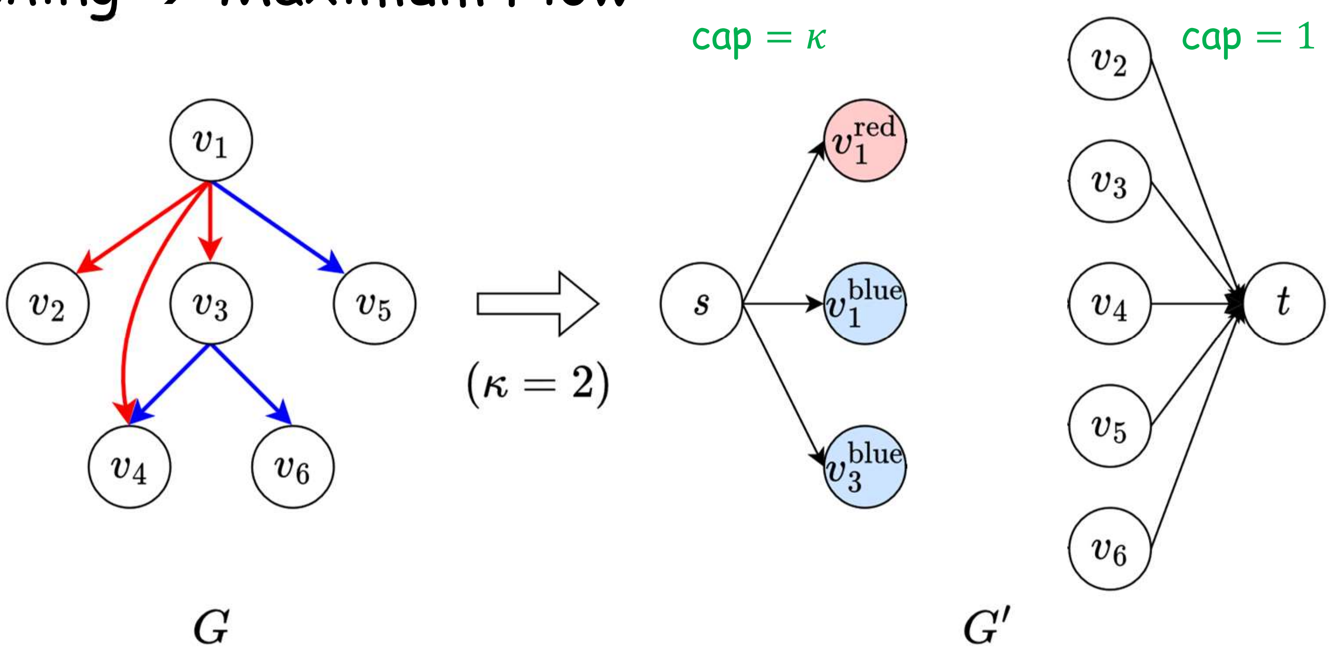
κ-COCST on λ-edge-colored DAGs

- Match each non-root vertex to one of its incoming edge, such that colored-out-constraint holds
- B-matching → Maximum Flow



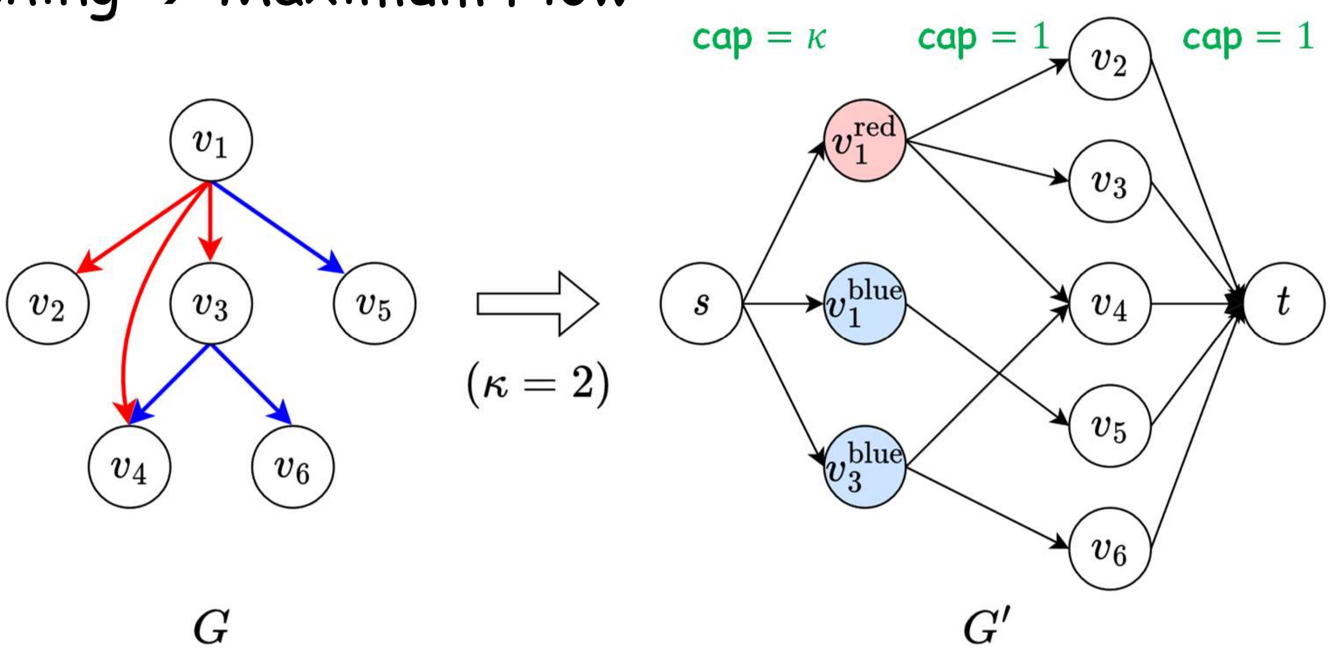
κ-COCST on λ-edge-colored DAGs

- Match each non-root vertex to one of its incoming edge, such that colored-out-constraint holds
- B-matching → Maximum Flow



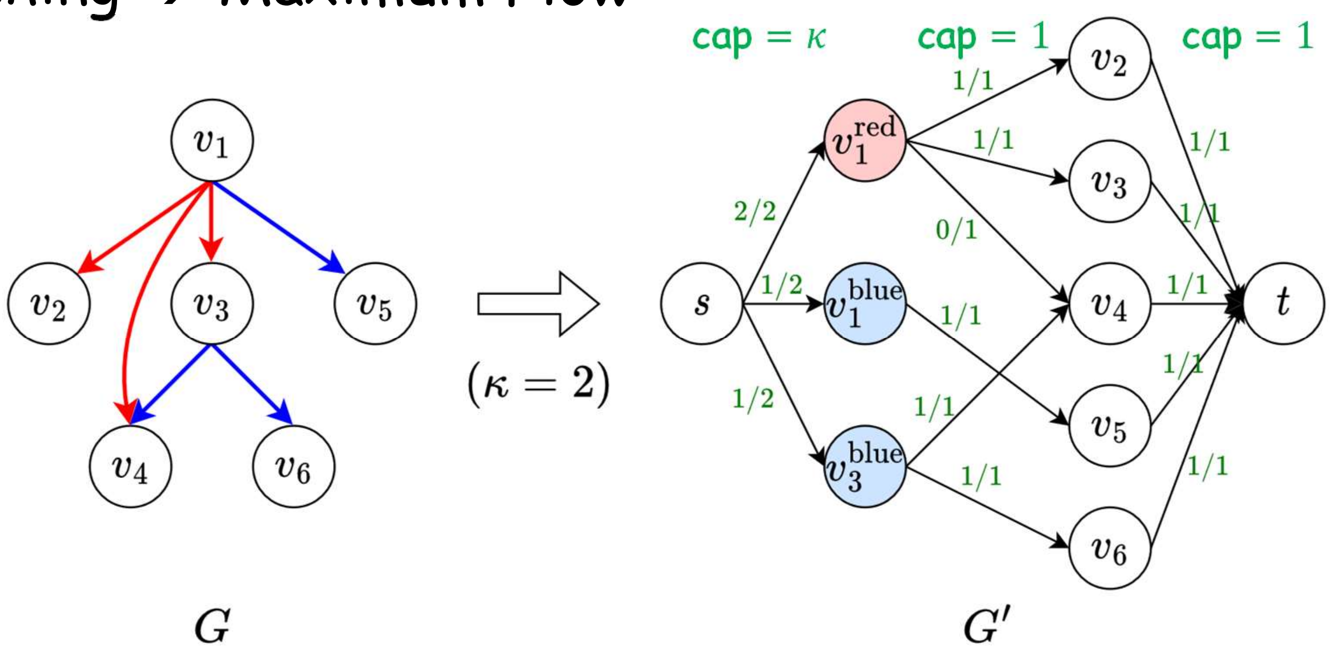
κ-COCST on λ-edge-colored DAGs

- Match each non-root vertex to one of its incoming edge, such that colored-out-constraint holds
- B-matching → Maximum Flow



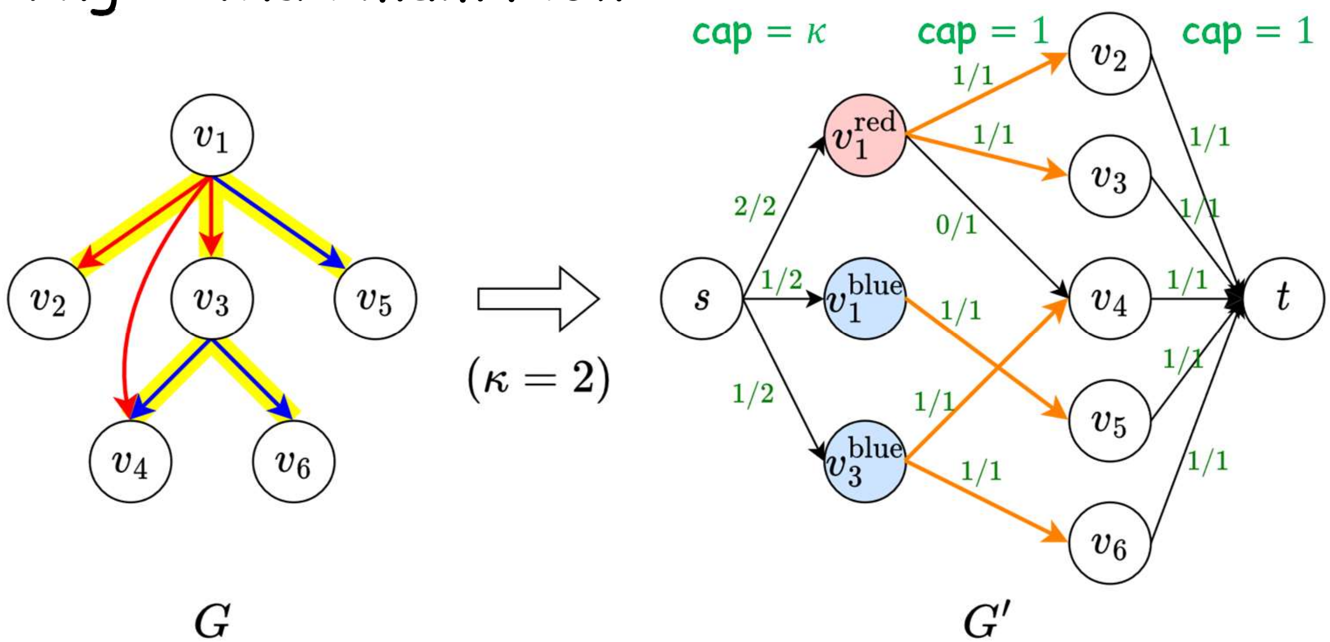
κ-COCST on λ-edge-colored DAGs

- Match each non-root vertex to one of its incoming edge, such that colored-out-constraint holds
- B-matching → Maximum Flow



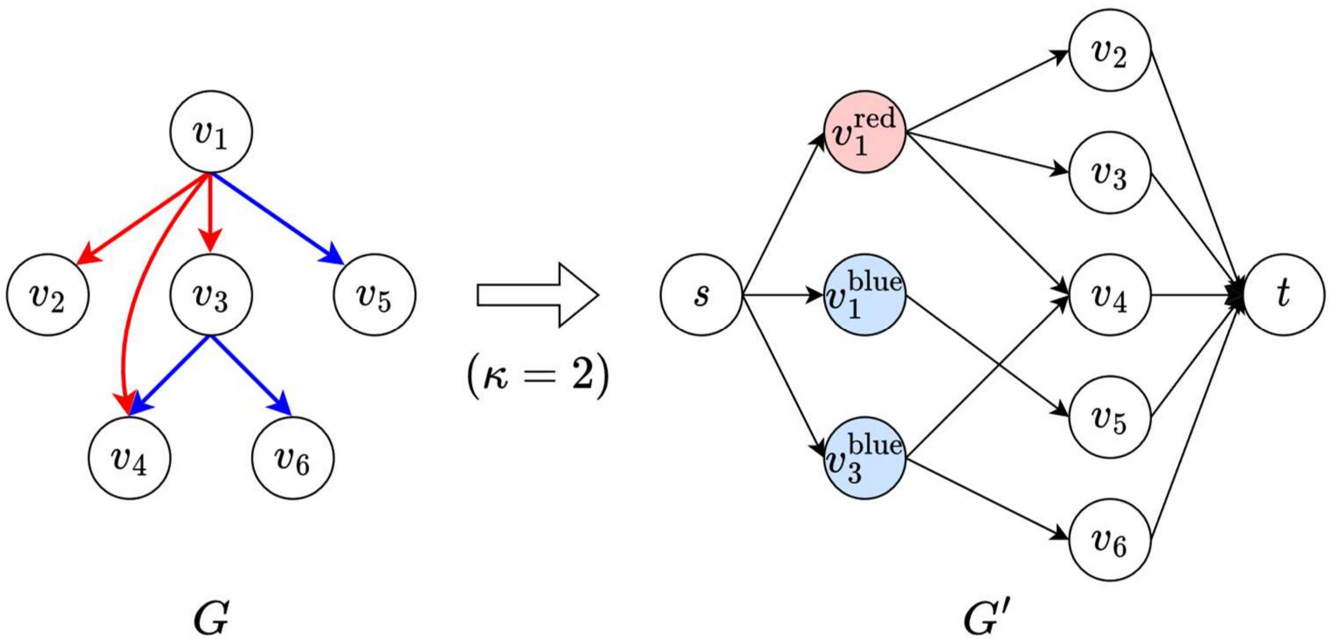
κ-COCST on λ-edge-colored DAGs

- Match each non-root vertex to one of its incoming edge, such that colored-out-constraint holds
- B-matching → Maximum Flow



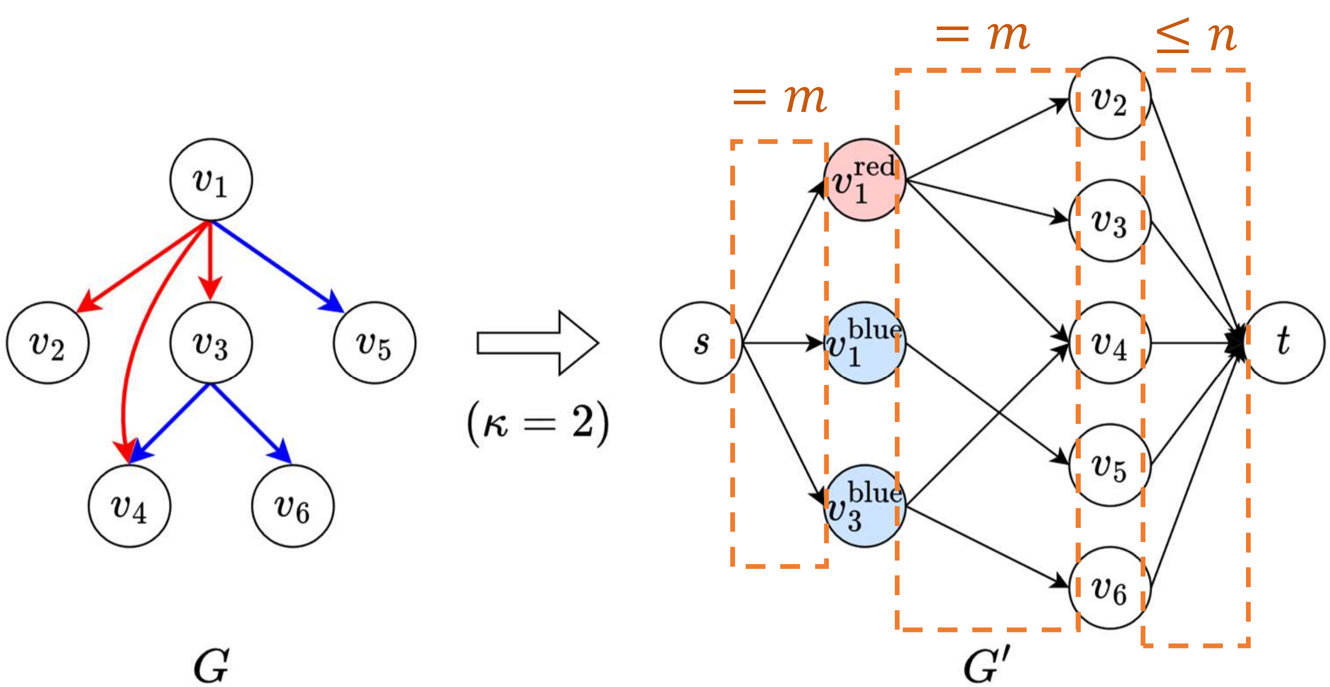
κ-COCST on λ-edge-colored DAGs

- Suppose the edge-colored graph has m edges and n vertices
- Flow graph has at most $2m + n$ edges and $m + n - 1$ vertices



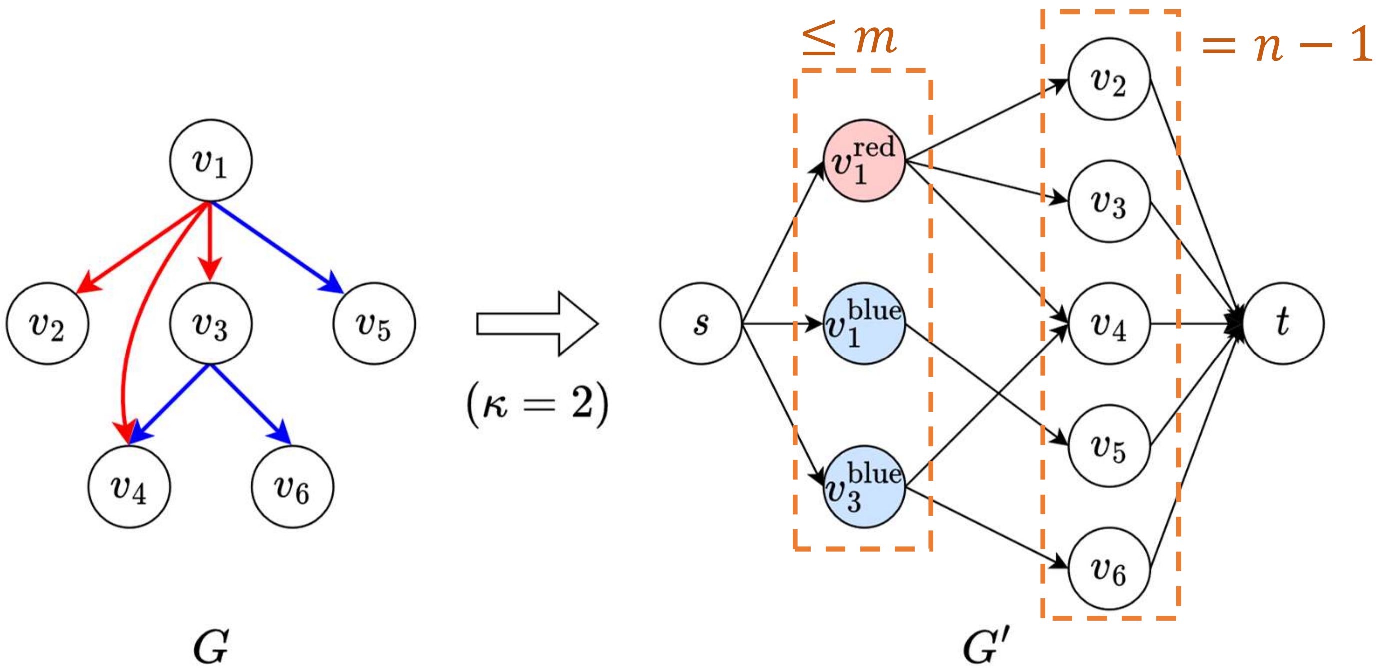
κ-COCST on λ-edge-colored DAGs

- Suppose the edge-colored graph has m edges and n vertices
- Flow graph has at most $2m + n$ edges and $m + n - 1$ vertices



κ-COCST on λ-edge-colored DAGs

- Suppose the edge-colored graph has m edges and n vertices
- Flow graph has at most $2m + n$ edges and $m + n - 1$ vertices



Hardness results

κ-CCST on DAGs

Borozan et al. (Eur. J. Comb. 2019)

κ-CCST
on undirected graphs
NP-Hard

reduction →

κ-CCST
on directed graphs

κ-CCST on DAGs

Borozan et al. (Eur. J. Comb. 2019)

κ-CCST
on undirected graphs
NP-Hard

→ reduction →

κ-CCST
on directed graphs

degree-constrained
spanning tree
problem

→ reduction →

κ-CCST on DAGs

κ-CCST on DAGs

Borozan et al. (Eur. J. Comb. 2019)

κ-CCST
on undirected graphs
NP-Hard

reduction

κ-CCST
on directed graphs

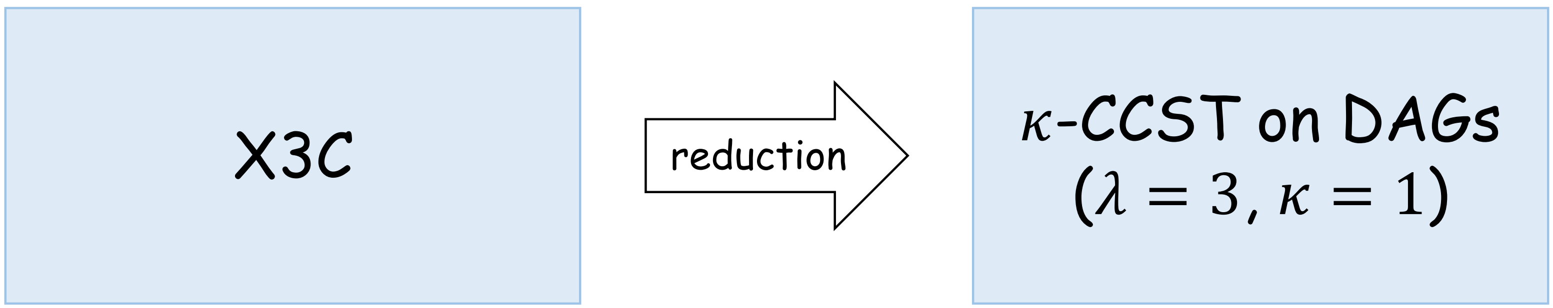
degree-constrained
spanning tree
problem

reduction

κ-CCST on DAGs

polynomial time solvable on DAGs

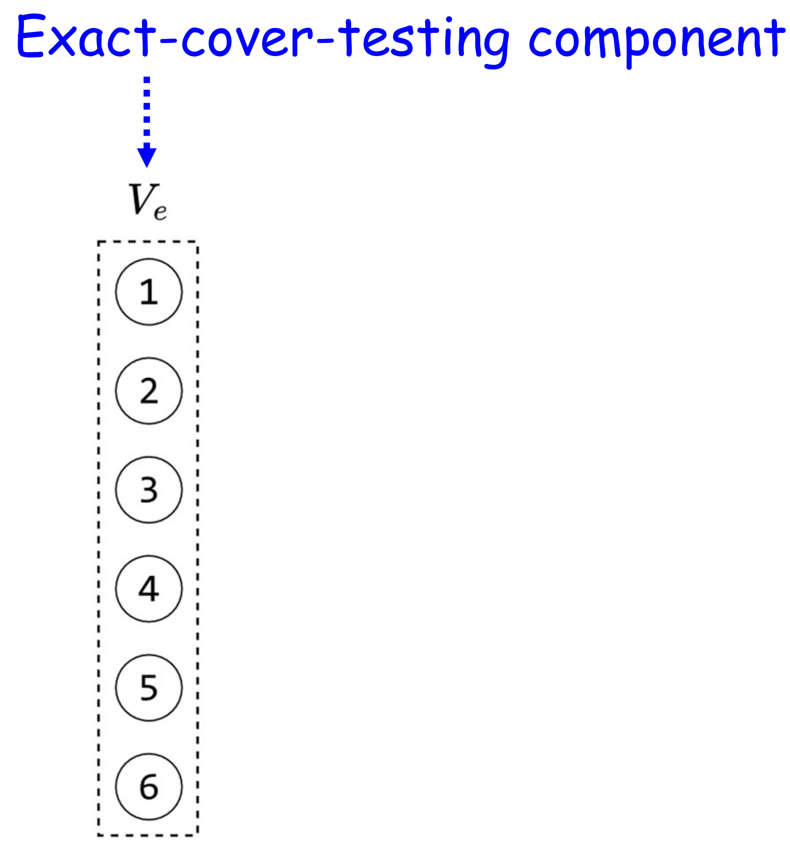
κ-CCST on DAGs



- NP-Hard problem X3C:
  - Input: $X = \{x_1, x_2, \dots, x_{3n}\}$ and $\mathrm{C} = \{c_1, c_2, \dots, c_m\}$.
  - Output: Are there n elements of C whose union is X?
  - Ex: $(m = 5, n = 2)$
    $X = \{1, 2, 3, 4, 5, 6\}$,
    $C = \{\{1,2,3\}, \{2,3,4\}, \{1,2,5\}, \{2,5,6\}, \{1,5,6\}\}$.

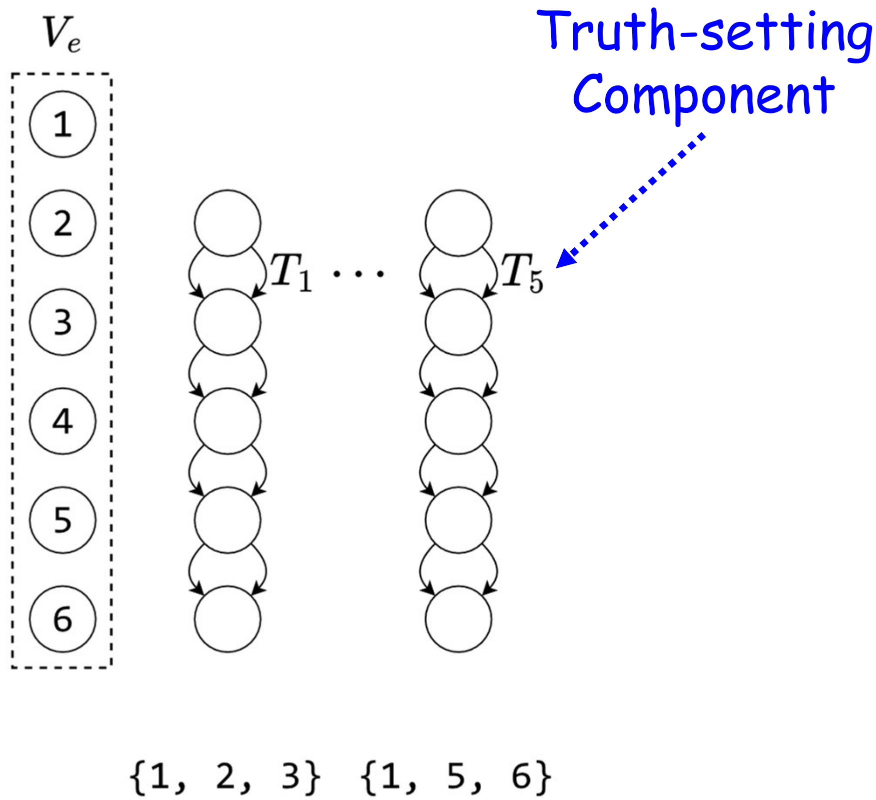
κ-CCST on DAGs

- Ex: $(m = 5, n = 2)$
  $X = \{1, 2, 3, 4, 5, 6\}$
  $C = \{\{1,2,3\}, \dots, \{1,5,6\}\}$



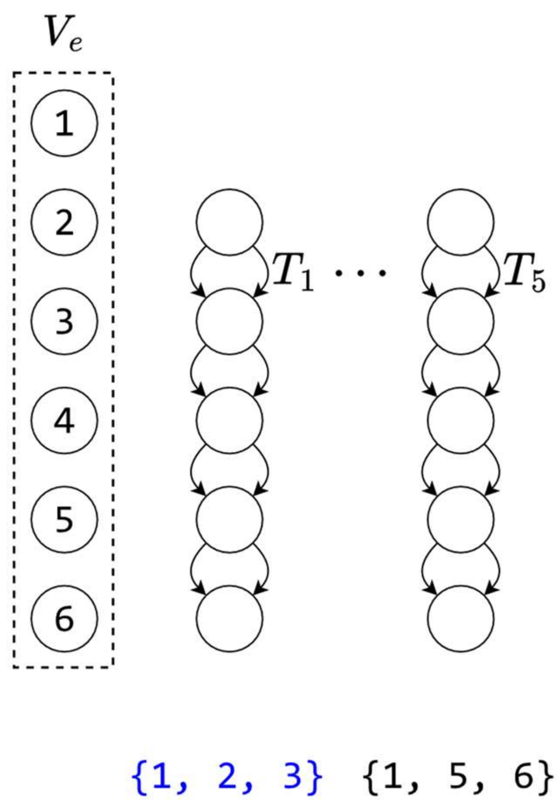
κ-CCST on DAGs

- Ex: $(m = 5, n = 2)$
  $X = \{1, 2, 3, 4, 5, 6\}$
  $C = \{\{1,2,3\}, \dots, \{1,5,6\}\}$



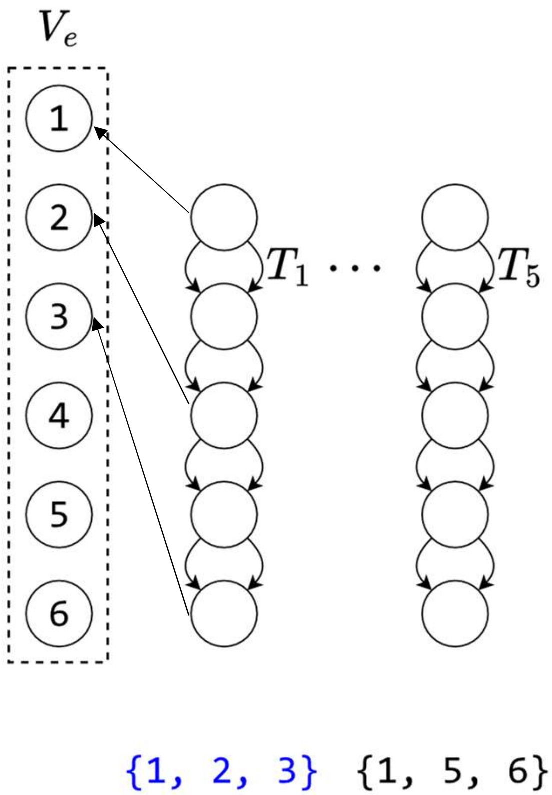
κ-CCST on DAGs

- Ex: $(m = 5, n = 2)$
  $X = \{1, 2, 3, 4, 5, 6\}$
  $C = \{\{1,2,3\}, \dots, \{1,5,6\}\}$



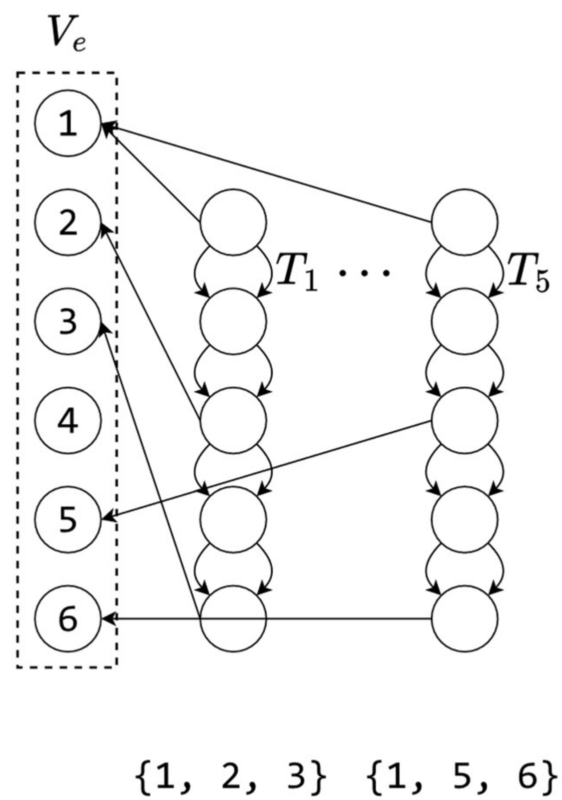
κ-CCST on DAGs

- Ex: $(m = 5, n = 2)$
  $X = \{1, 2, 3, 4, 5, 6\}$
  $C = \{\{1,2,3\}, \ldots, \{1,5,6\}\}$



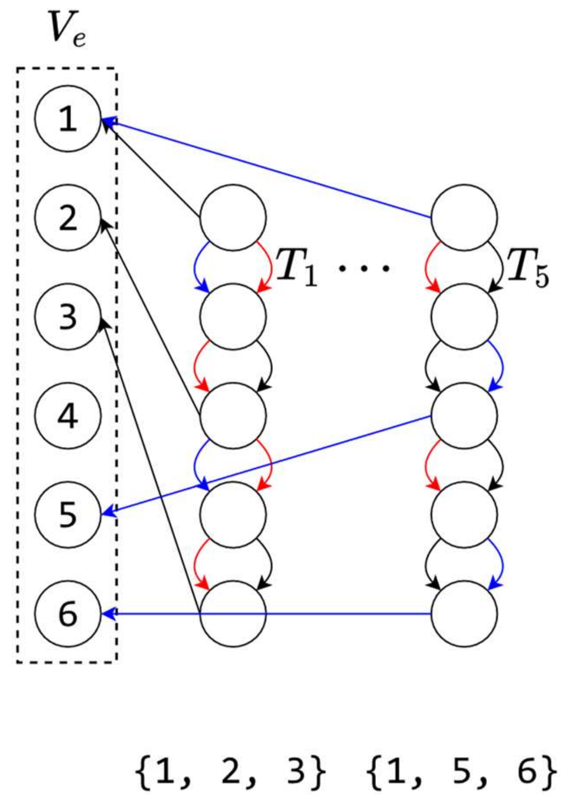
κ-CCST on DAGs

- Ex: $(m = 5, n = 2)$
  $X = \{1, 2, 3, 4, 5, 6\}$
  $C = \{\{1,2,3\}, \dots, \{1,5,6\}\}$



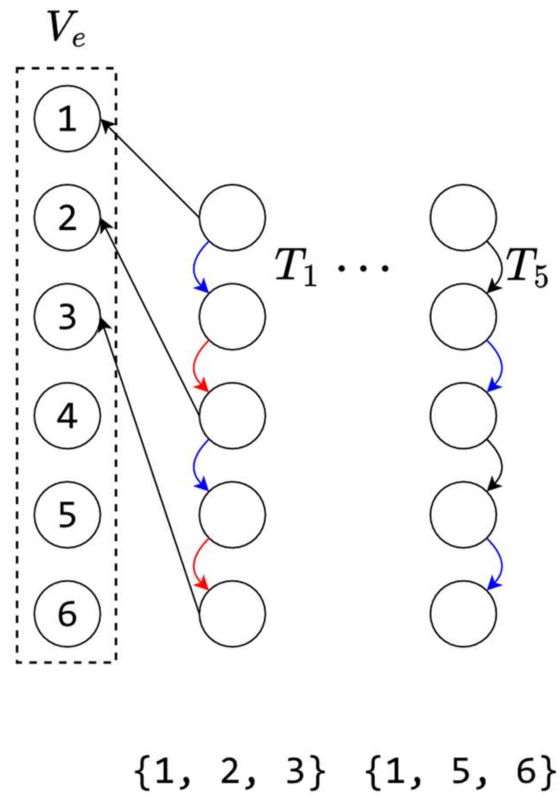
κ-CCST on DAGs

- Ex: $(m = 5, n = 2)$
  $X = \{1, 2, 3, 4, 5, 6\}$
  $C = \{\{1,2,3\}, \dots, \{1,5,6\}\}$



κ-CCST on DAGs

- Ex: $(m = 5, n = 2)$
  $X = \{1, 2, 3, 4, 5, 6\}$
  $C = \{\{1,2,3\}, \dots, \{1,5,6\}\}$



chosen v.s. not chosen

κ-COCST on directed graphs

κ-CCST
on directed graphs
$(\lambda \geq 2, \kappa \geq 1)$

reduction

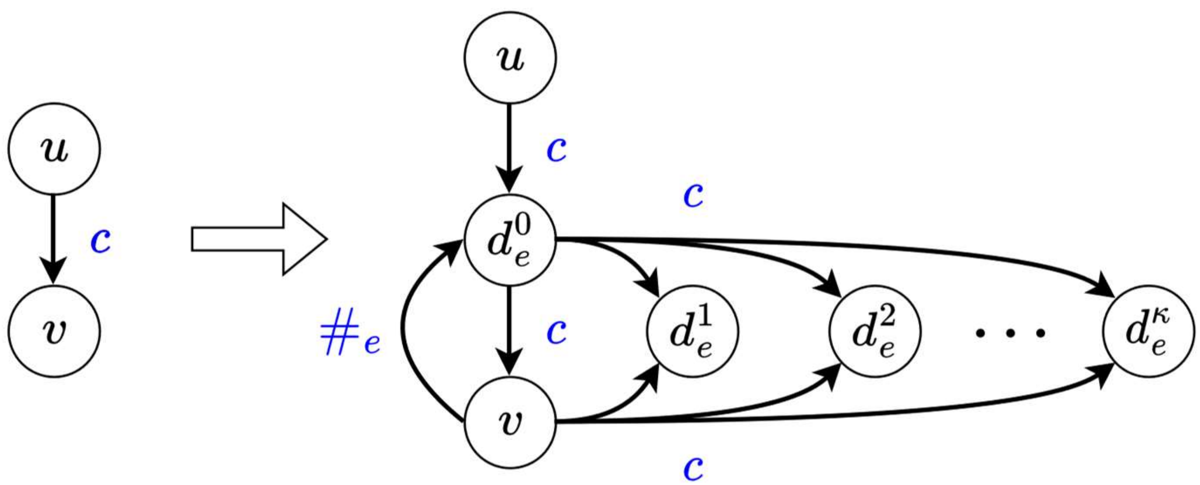
κ-COCST
on directed graphs
$(\lambda \geq 4, \kappa \geq 1)$

NP-Hard

κ-COCST on directed graphs

- Replace each edge $e = (u, v, c)$ with a gadget G_e.

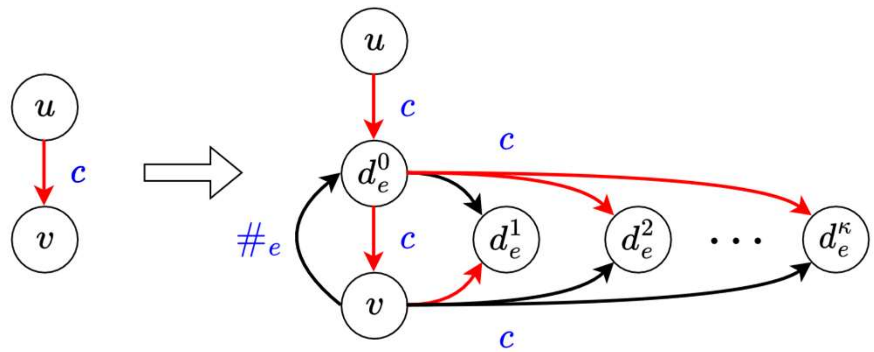
edge e



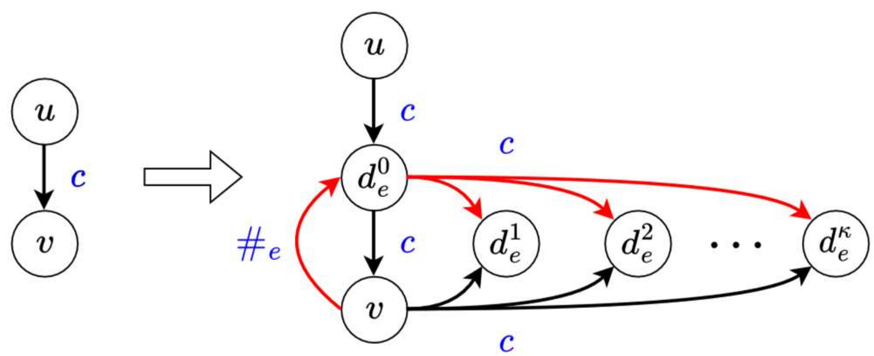
gadget G_e

κ-COCST on directed graphs

chosen



not chosen



Conclusion

- κ-CCST

Graph Class	λ	κ	Hardness
Directed	≥ 2	≥ 1	NP-Hard
DAG	≥ 3	≥ 1	NP-Hard
DAG	$= 2$	≥ 2	Open
DAG	$= 2$	$= 1$	P

- κ-COCST

Graph Class	λ	κ	Hardness
Directed	≥ 4	≥ 1	NP-Hard
Directed	$= 3$	$= 1$	Open
Directed	$= 2$	≥ 1	Open
DAG	≥ 2	≥ 1	P