The Parameterized Complexity of Coloring Mixed Graphs

SWAT 2026

Konstanty Junosza-Szaniawski

Antonio Lauerbach

Marie Diana Sieper

Alexander Wolff

Julius-Maximilians-UNIVERSITÄT WÜRZBURG

Warsaw University of Technology

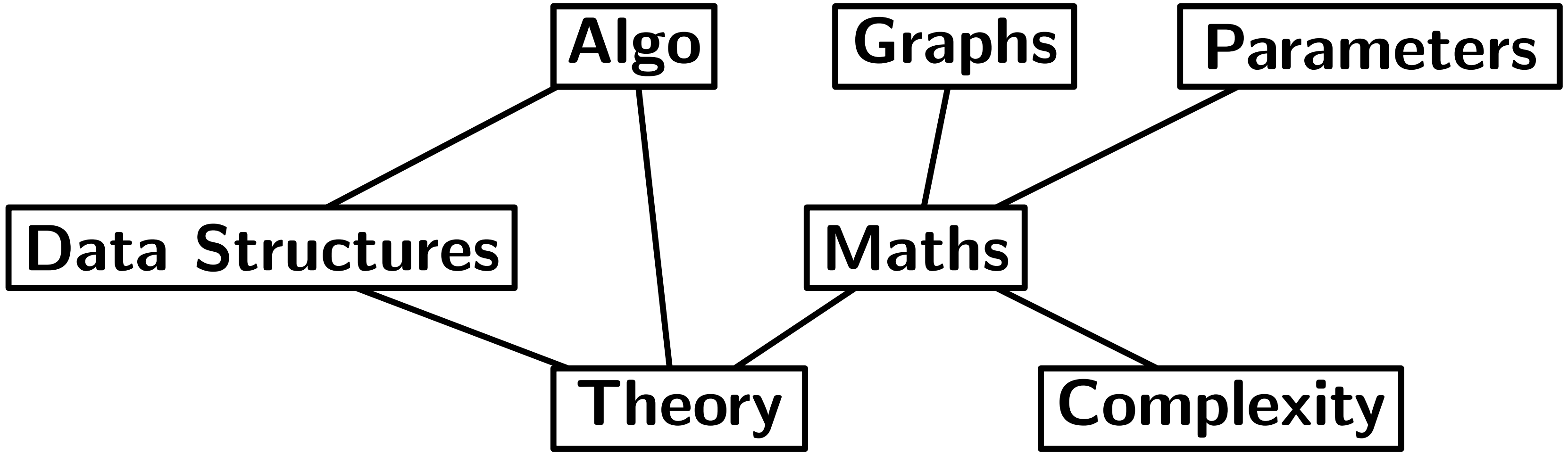
Introduction

Algo **Graphs** **Parameters**

Data Structures **Maths**

Theory **Complexity**

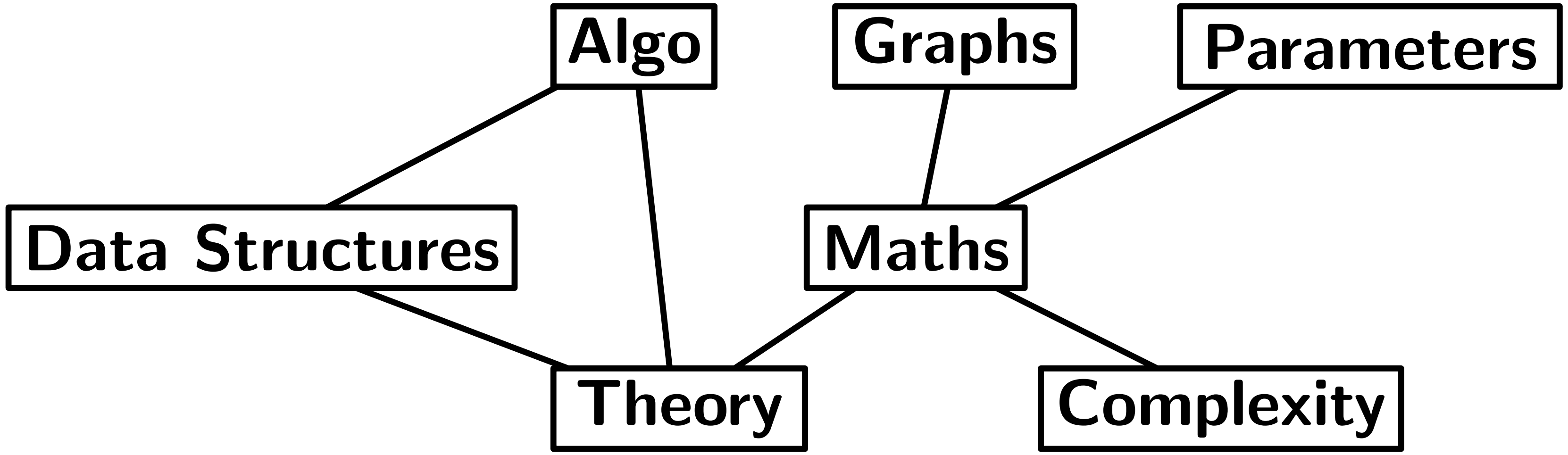
Introduction



Introduction

Undirected Graph

- vertices V
- edges E



Introduction

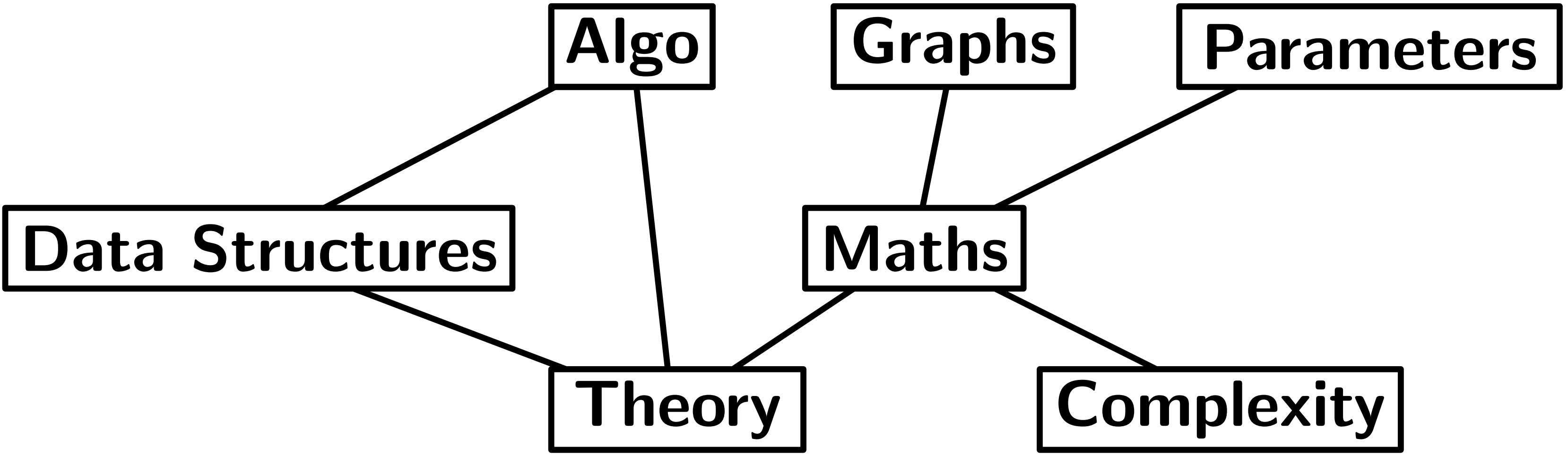
Undirected Graph

- vertices V
- edges E

Coloring

k-coloring $c\colon V \to [k]$ proper if

- $\forall \{u, v\} \in E\colon c(u) \neq c(v)$



Introduction

Undirected Graph

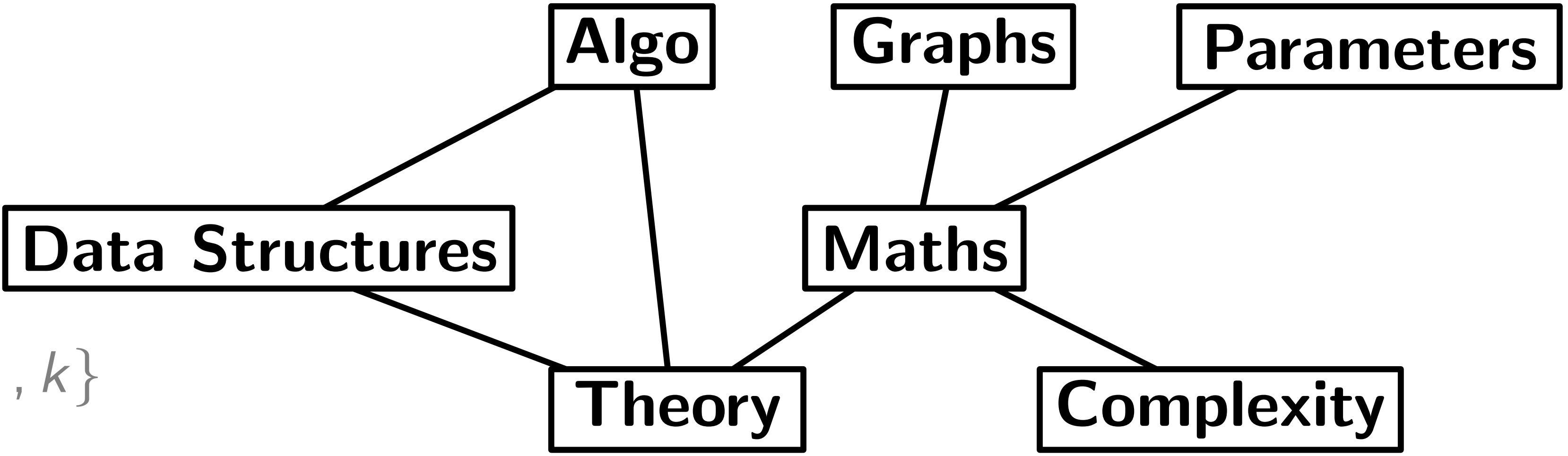
- vertices V
- edges E

Coloring

$\{1, \ldots, k\}$

k-coloring $c \colon V \to [k]$ proper if

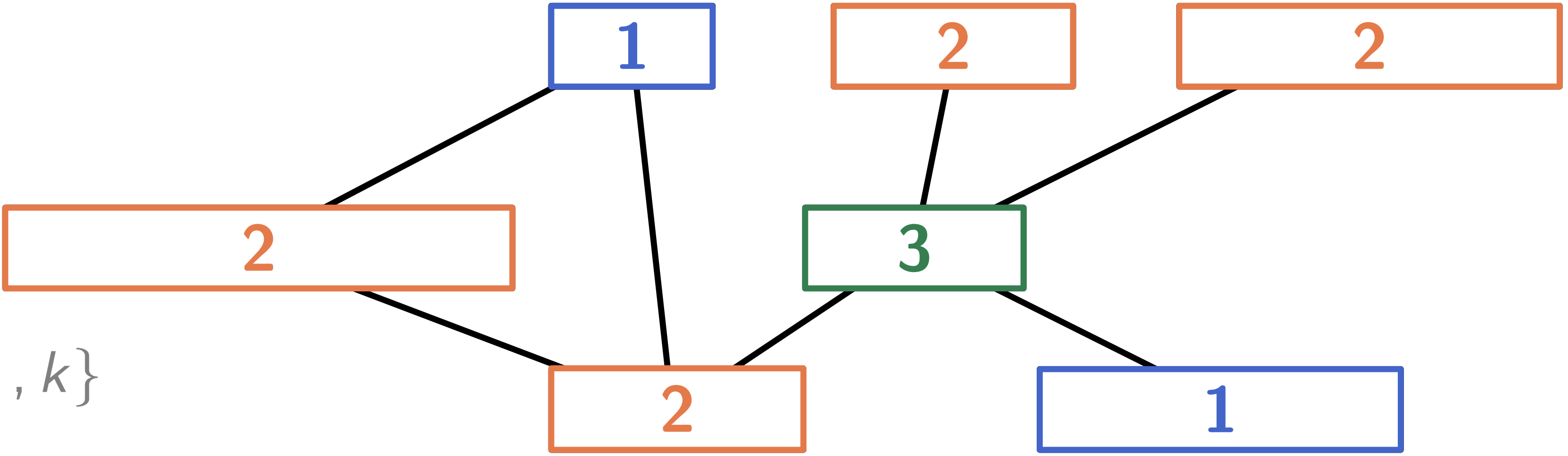
- $\forall \{u, v\} \in E \colon c(u) \neq c(v)$



Introduction

Undirected Graph

- ♦ vertices V
- ♦ edges E



Coloring

k-coloring $c\colon V \to [k]$ proper if

$\{1, \ldots, k\}$

- ♦ $\forall \{u, v\} \in E\colon c(u) \neq c(v)$

Introduction

Undirected Graph

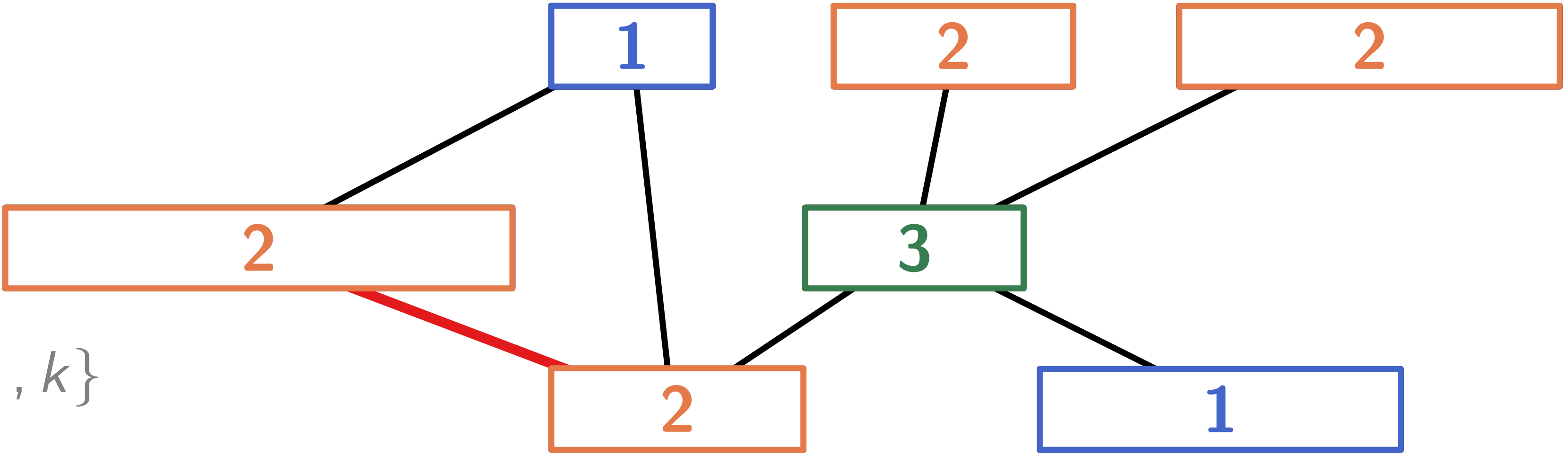
- vertices V
- edges E

Coloring

k-coloring $c\colon V \to [k]$ proper if

$\{1, \ldots, k\}$

- $\forall \{u, v\} \in E\colon c(u) \neq c(v)$



Introduction

Undirected Graph

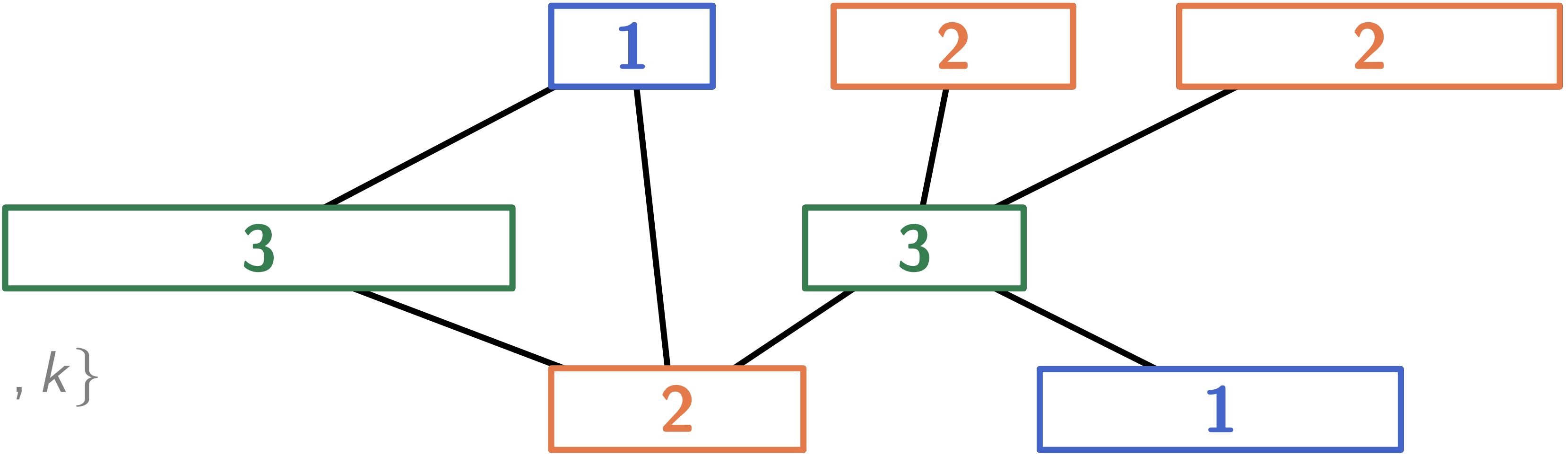
- vertices V
- edges E

Coloring

$\{1, \ldots, k\}$

k-coloring $c\colon V \to [k]$ proper if

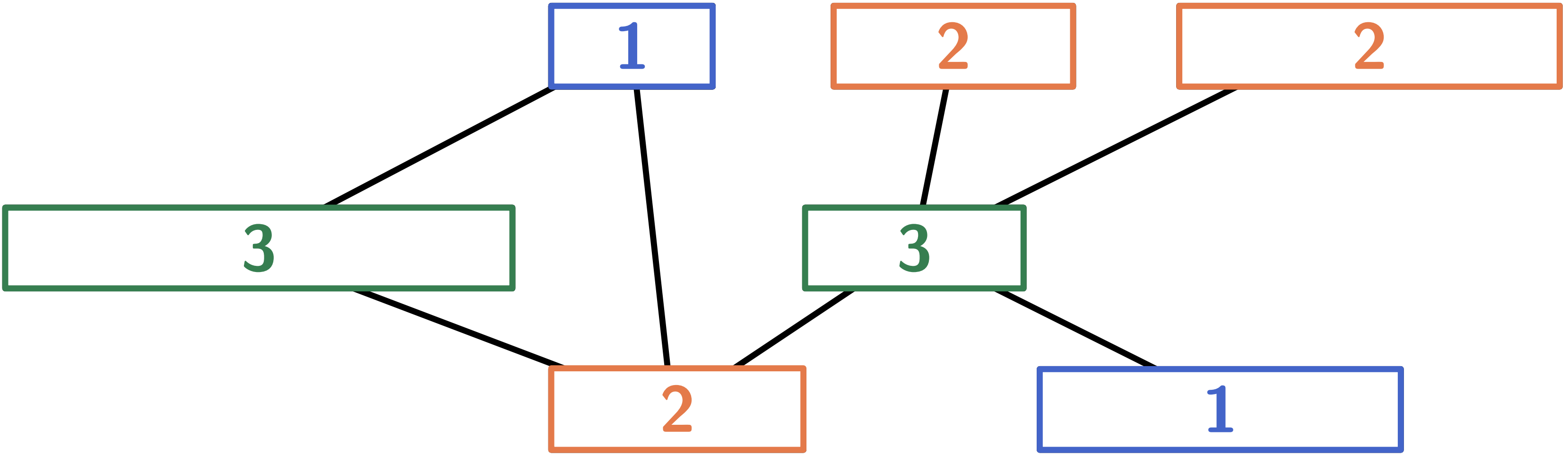
- $\forall \{u, v\} \in E\colon c(u) \neq c(v)$



Introduction

Undirected Graph

- ♦ vertices V
- ♦ edges E



Coloring

k-coloring $c\colon V \to [k]$ proper if

- ♦ $\forall \{u, v\} \in E\colon c(u) \neq c(v)$

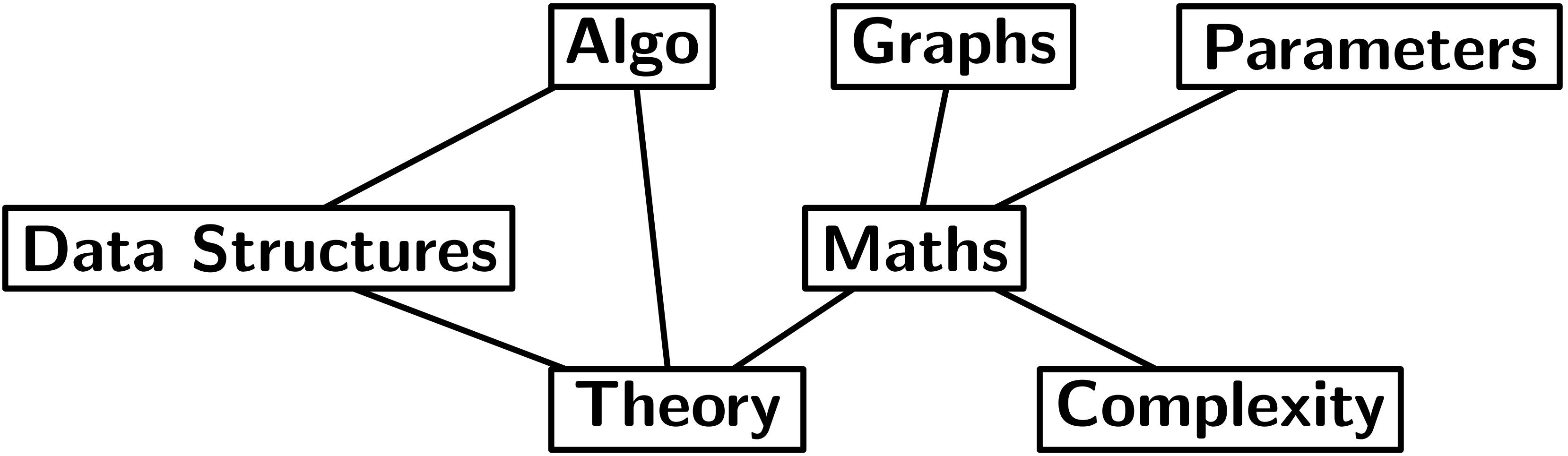
COLORING

Is there a proper k-coloring of a given graph G?

Introduction

Undirected Graph

- vertices V
- edges E



Coloring

k-coloring $c\colon V \to [k]$ proper if

- $\forall\{u,v\} \in E\colon c(u) \neq c(v)$

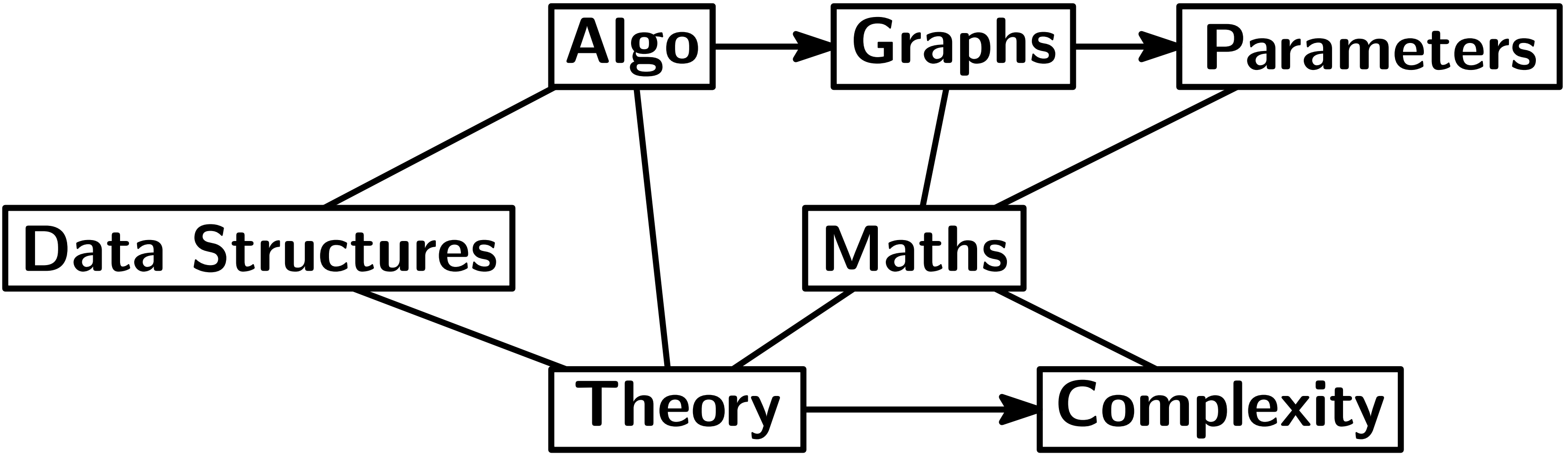
Coloring

Is there a proper k-coloring of a given graph G?

Introduction

Undirected Graph

- vertices V
- edges E



Coloring

k-coloring $c\colon V \to [k]$ proper if

- $\forall \{u, v\} \in E\colon c(u) \neq c(v)$

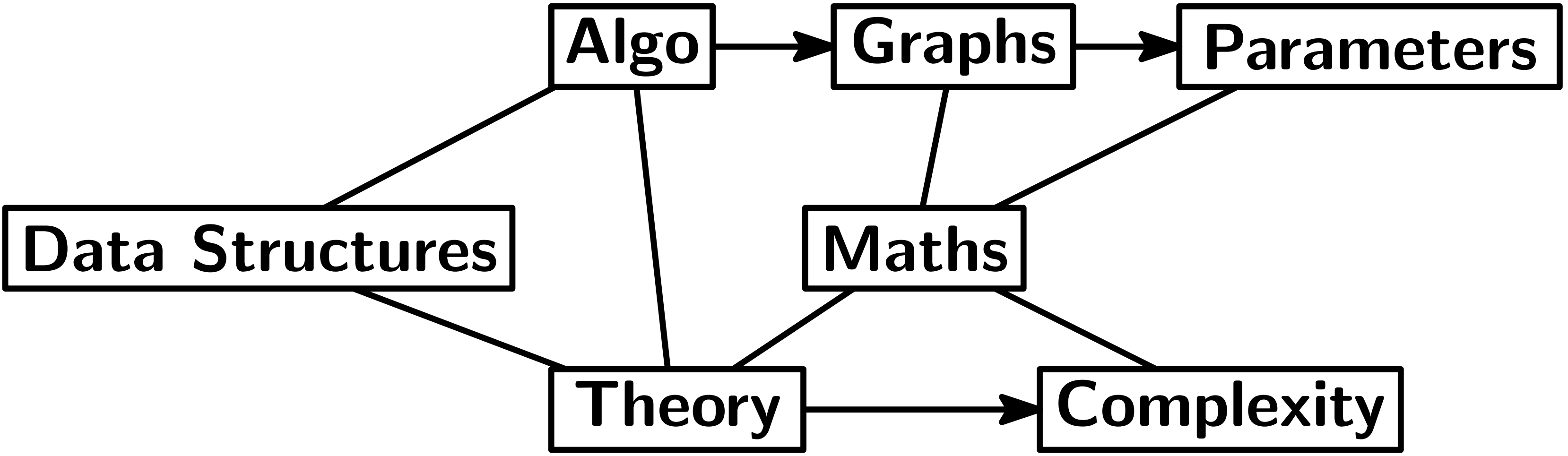
Coloring

Is there a proper k-coloring of a given graph G?

Introduction

Mixed Graph

- vertices V
- edges E
- arcs A



Coloring

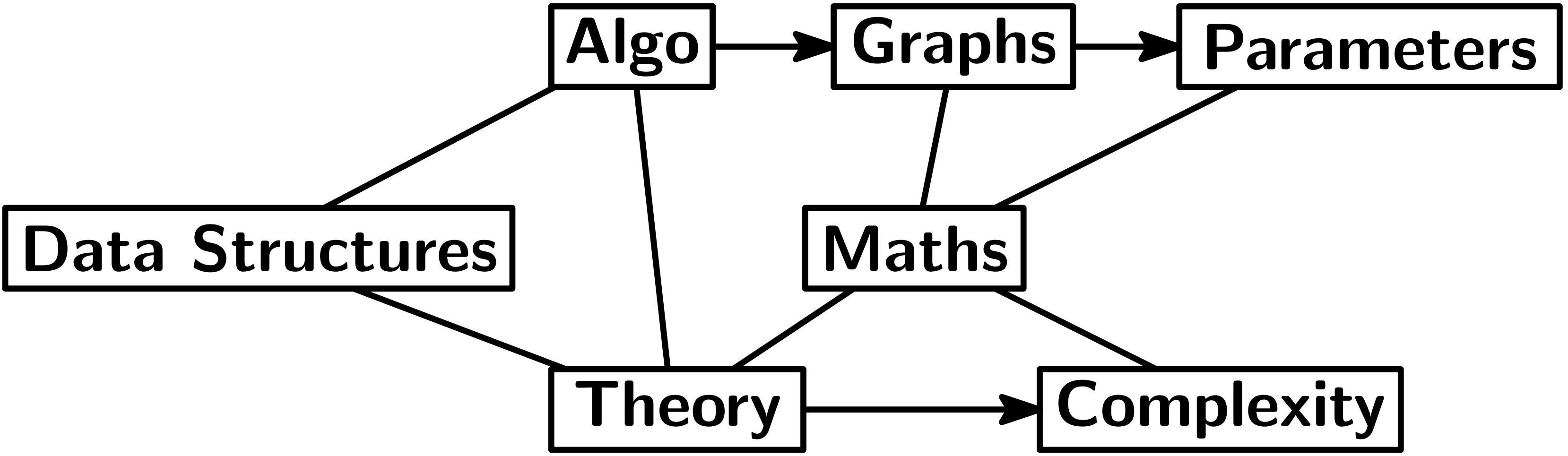
k-coloring $c\colon V \to [k]$ proper if

- $\forall \{u, v\} \in E\colon c(u) \neq c(v)$

COLORING

Is there a proper k-coloring of a given graph G?

Introduction



Mixed Graph

- vertices V
- edges E
- arcs A

Coloring

k-coloring $c\colon V \to [k]$ proper if

- $\forall \{u, v\} \in E\colon c(u) \neq c(v)$
- $\forall (u, v) \in A\colon c(u) < c(v)$

COLORING

Is there a proper k-coloring of a given graph G?

Introduction

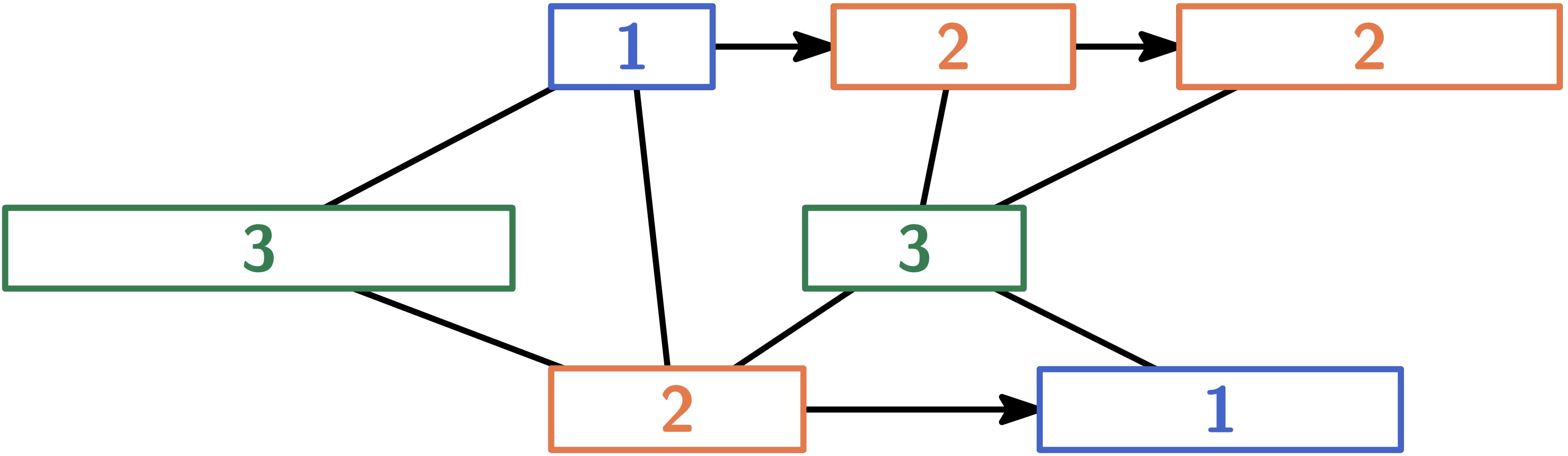
Mixed Graph

- vertices V
- edges E
- arcs A

Coloring

k-coloring $c\colon V \to [k]$ proper if

- $\forall \{u, v\} \in E\colon c(u) \neq c(v)$
- $\forall (u, v) \in A\colon c(u) < c(v)$



COLORING

Is there a proper k-coloring of a given graph G?

Introduction

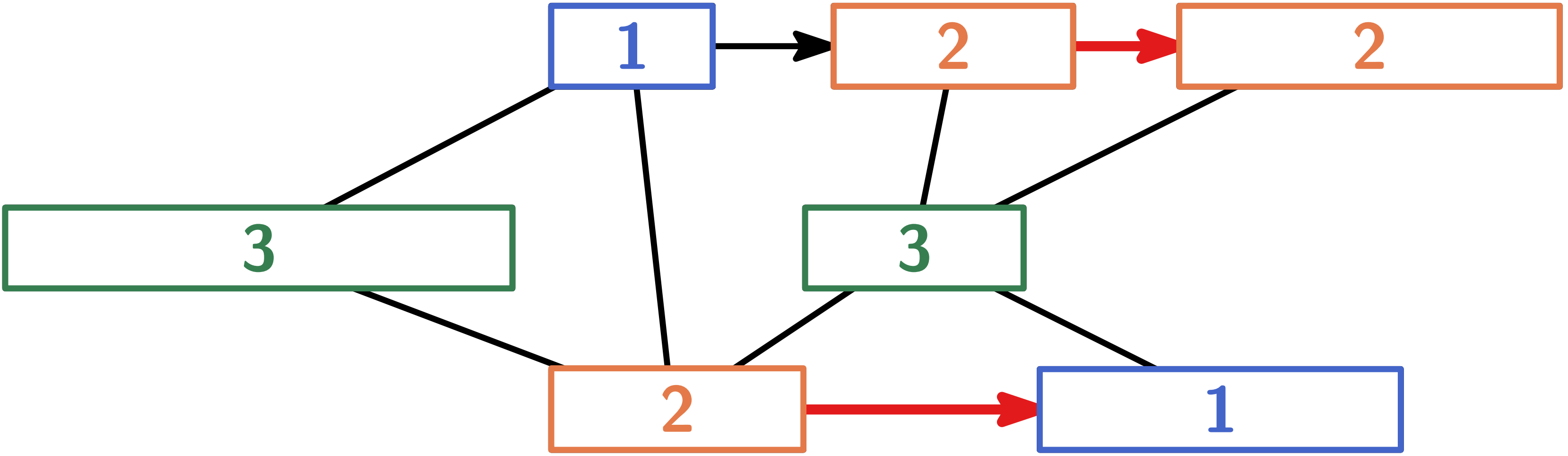
Mixed Graph

- vertices V
- edges E
- arcs A

Coloring

k-coloring $c\colon V \to [k]$ proper if

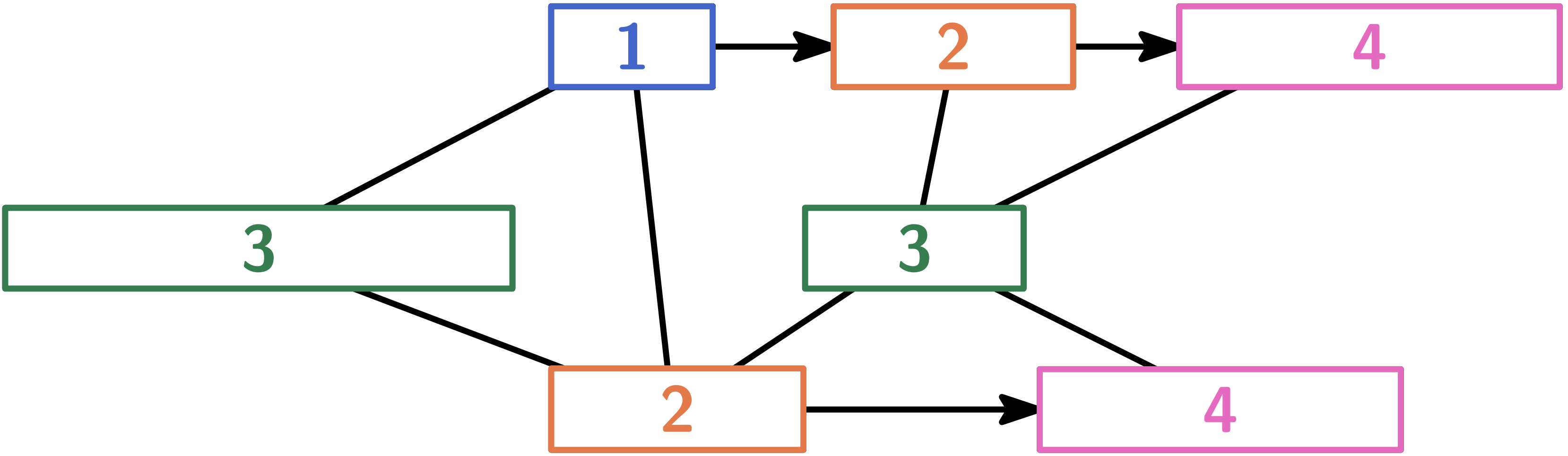
- $\forall \{u, v\} \in E\colon c(u) \neq c(v)$
- $\forall (u, v) \in A\colon c(u) < c(v)$



COLORING

Is there a proper k-coloring of a given graph G?

Introduction



Mixed Graph

- vertices V
- edges E
- arcs A

Coloring

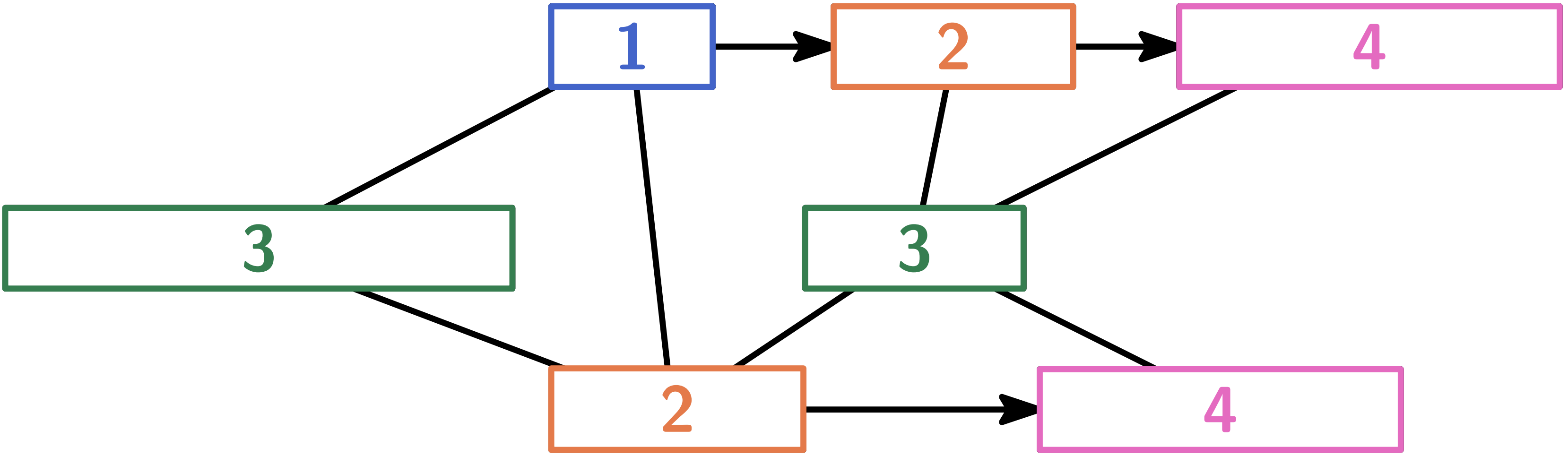
k-coloring $c\colon V \to [k]$ proper if

- $\forall \{u, v\} \in E\colon c(u) \neq c(v)$
- $\forall (u, v) \in A\colon c(u) < c(v)$

COLORING

Is there a proper k-coloring of a given graph G?

Introduction



Mixed Graph

- vertices V
- edges E
- arcs A

Coloring

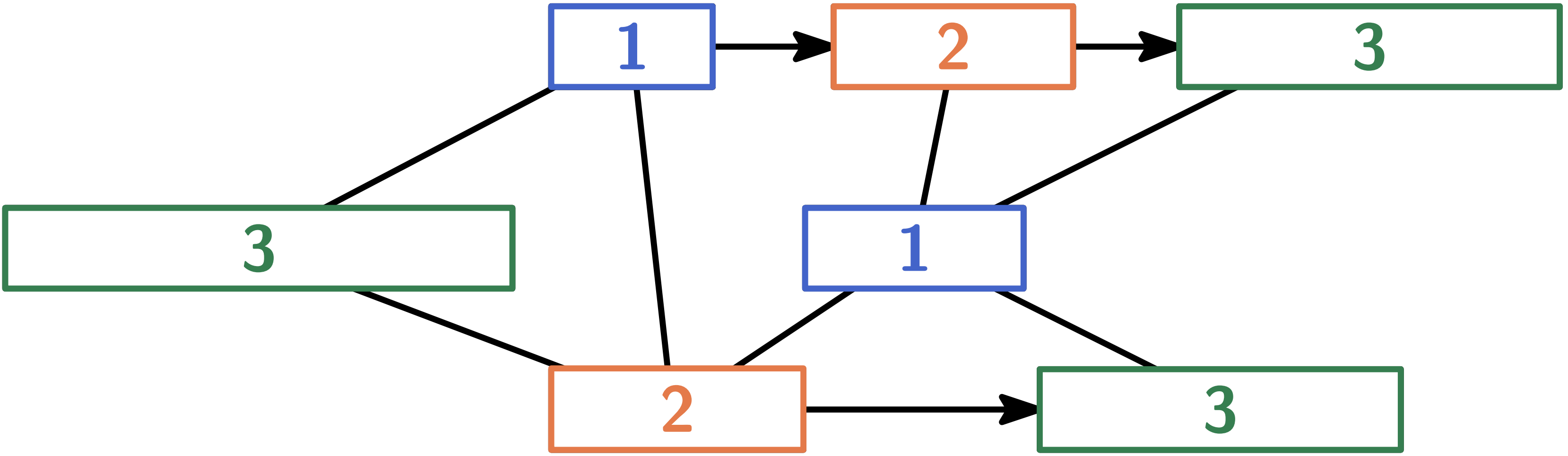
k-coloring $c\colon V \to [k]$ proper if

- $\forall \{u, v\} \in E\colon c(u) \neq c(v)$
- $\forall (u, v) \in A\colon c(u) < c(v)$

MIXEDCOLORING

Is there a proper k-coloring of a given mixed graph G?

Introduction



Mixed Graph

- vertices V
- edges E
- arcs A

Coloring

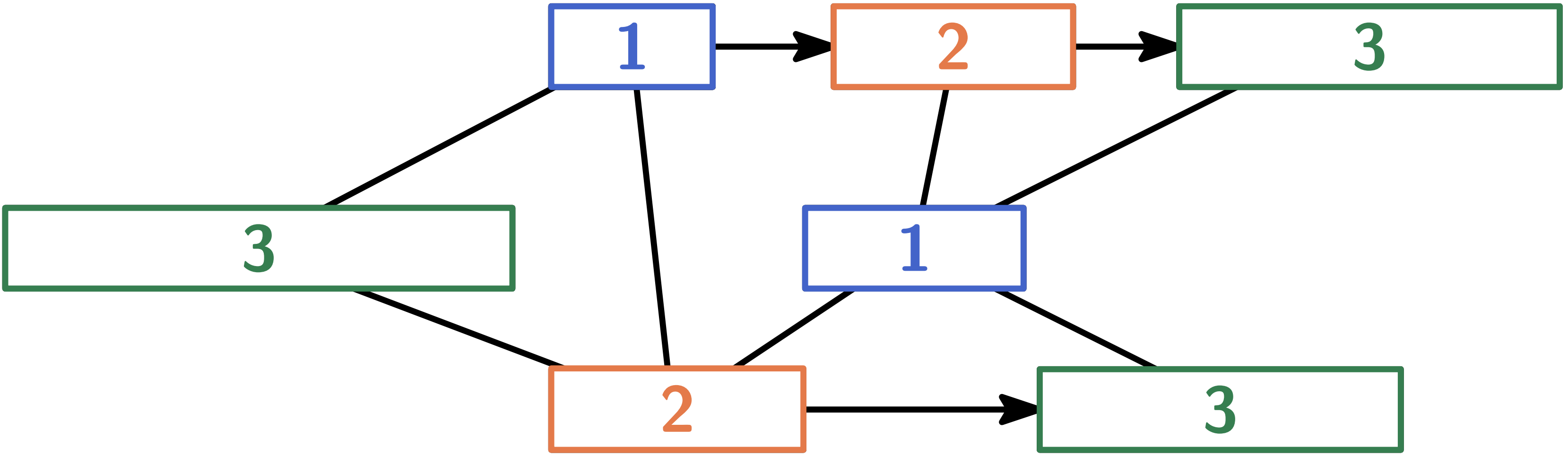
k-coloring $c\colon V \to [k]$ proper if

- $\forall \{u, v\} \in E\colon c(u) \neq c(v)$
- $\forall (u, v) \in A\colon c(u) < c(v)$

MIXEDCOLORING

Is there a proper k-coloring of a given mixed graph G?

Introduction



Mixed Graph

- vertices V
- edges E
- arcs A

Coloring

k-coloring $c\colon V \to [k]$ proper if

- $\forall \{u,v\} \in E\colon c(u) \neq c(v)$
- $\forall (u,v) \in A\colon c(u) < c(v)$

MIXEDCOLORING

Is there a proper k-coloring of a given mixed graph G?

Applications

scheduling, graph drawing

Related Work

MixedColoring

Related Work

MixedColoring

- focus on special cases

Related Work

MixedColoring

- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics

Related Work

MixedColoring

- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics
- few results on parameterized complexity

Related Work

MixedColoring

- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics
- few results on parameterized complexity
  - FPT by #edges [Hansen et al. 97]

Related Work

MixedColoring

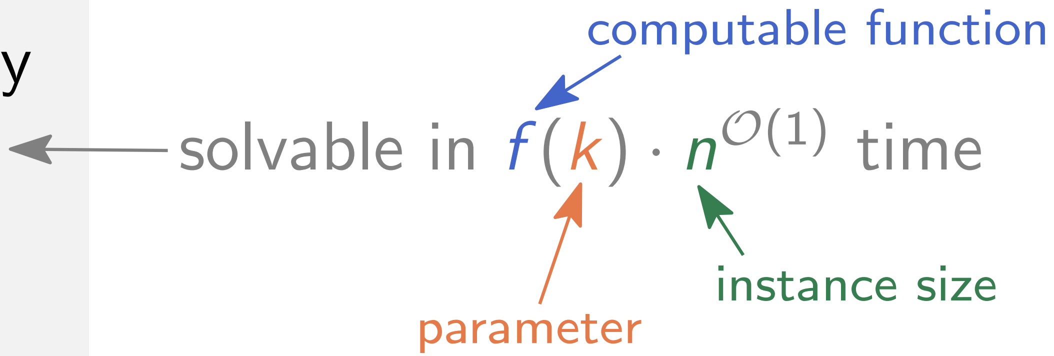
- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics
- few results on parameterized complexity
  - FPT by #edges [Hansen et al. 97]

← solvable in $f(k) \cdot n^{\mathcal{O}(1)}$ time

Related Work

MixedColoring

- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics
- few results on parameterized complexity
  - FPT by #edges [Hansen et al. 97]



computable function

solvable in $f(k) \cdot n^{\mathcal{O}(1)}$ time

parameter

instance size

Related Work

MixedColoring

- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics
- few results on parameterized complexity
  - ◆ FPT by #edges [Hansen et al. 97]

← solvable in $f(k) \cdot n^{\mathcal{O}(1)}$ time

Related Work

MIXEDCOLORING

- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics
- few results on parameterized complexity
  - FPT by #edges [Hansen et al. 97]
  - XP by treewidth [Ries & de Werra 08]

← solvable in $f(k) \cdot n^{\mathcal{O}(1)}$ time

Related Work

MixedColoring

- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics
- few results on parameterized complexity
  - FPT by #edges [Hansen et al. 97]
  - XP by treewidth [Ries & de Werra 08]

← solvable in $f(k) \cdot n^{\mathcal{O}(1)}$ time

← solvable in $\mathcal{O}(n^{f(k)})$ time

Related Work

MixedColoring

- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics
- few results on parameterized complexity
  - ♦ FPT by #edges [Hansen et al. 97]
  - ♦ XP by treewidth [Ries & de Werra 08]

solvable in $f(k) \cdot n^{\mathcal{O}(1)}$ time

solvable in $\mathcal{O}(n^{f(k)})$ time

considered w.r.t. underlying undirected graph

Related Work

MixedColoring

- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics
- few results on parameterized complexity
  - FPT by #edges [Hansen et al. 97] ← solvable in $f(k) \cdot n^{\mathcal{O}(1)}$ time
  - XP by treewidth [Ries & de Werra 08] ← solvable in $\mathcal{O}(n^{f(k)})$ time

considered w.r.t. underlying undirected graph

treats arcs as edges

Related Work

MixedColoring

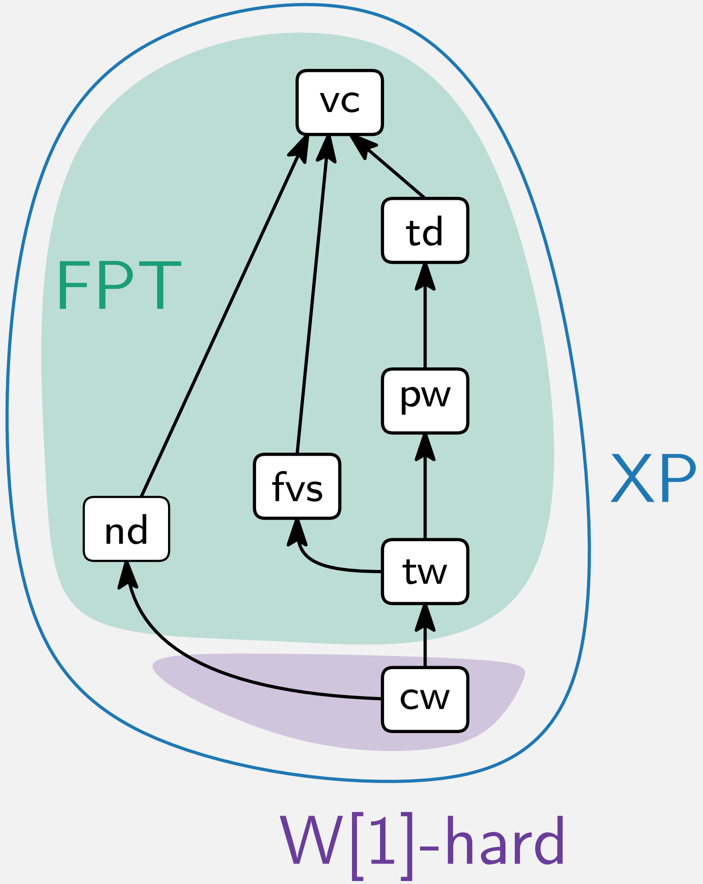
- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics
- few results on parameterized complexity
  - FPT by #edges [Hansen et al. 97]
  - XP by treewidth [Ries & de Werra 08]
  - hardness generalizes from Coloring

Related Work

MixedColoring

- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics
- few results on parameterized complexity
  - FPT by #edges [Hansen et al. 97]
  - XP by treewidth [Ries & de Werra 08]
  - hardness generalizes from Coloring

Coloring

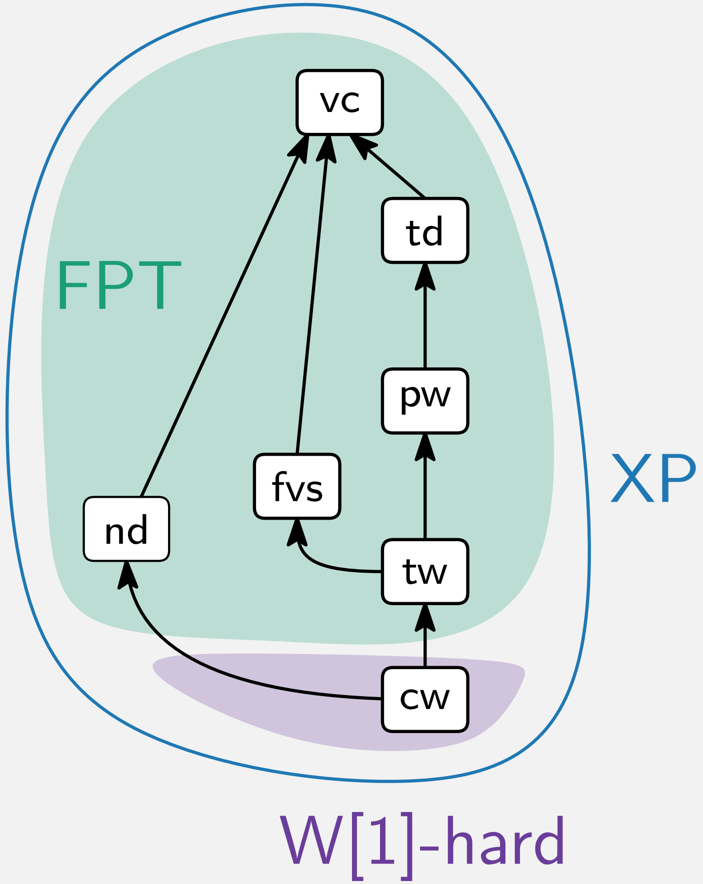


Related Work

MixedColoring

- focus on special cases
- algorithmic approaches: MILPs, branch-and-bound, and heuristics
- few results on parameterized complexity
  - FPT by #edges [Hansen et al. 97]
  - XP by treewidth [Ries & de Werra 08]
  - hardness generalizes from Coloring

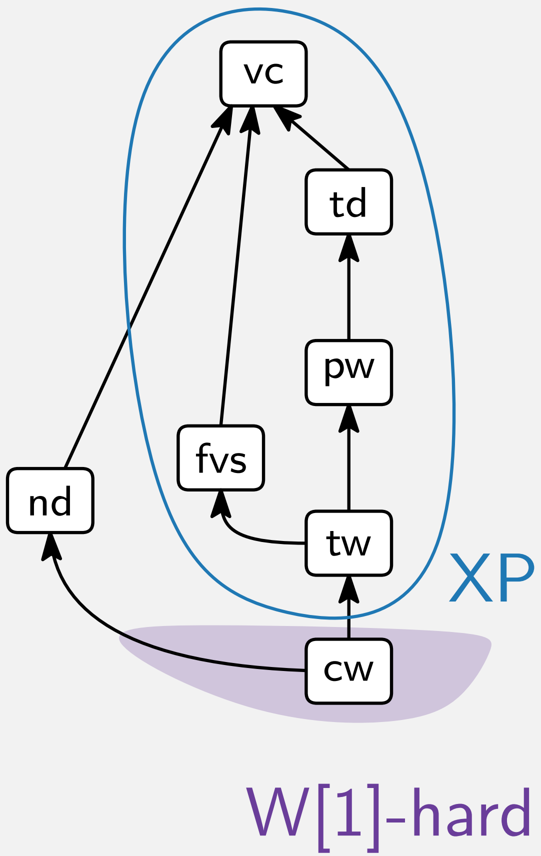
Coloring



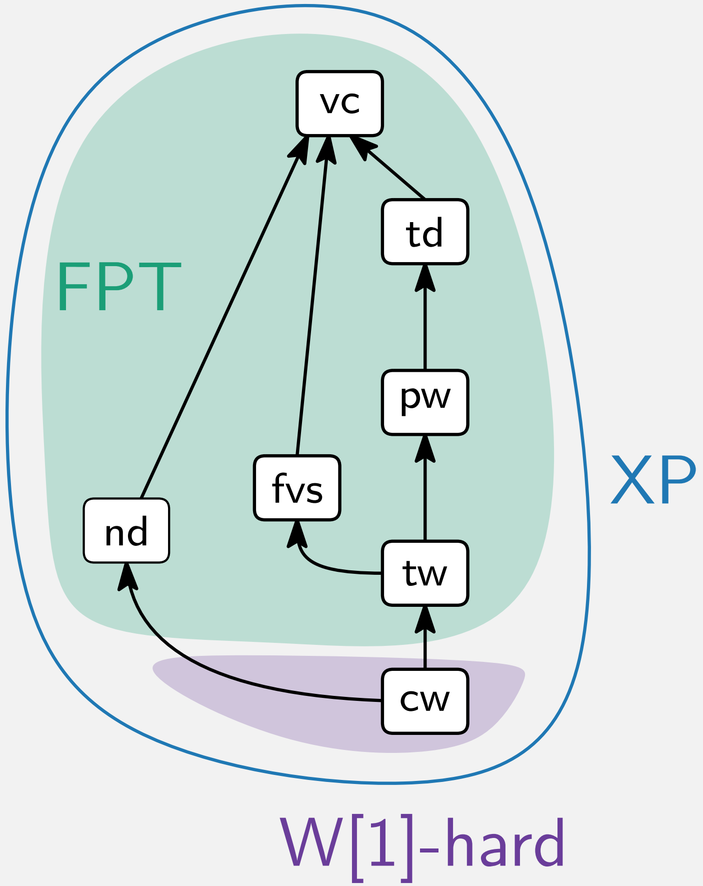
to FPT what NP-hardness is to P

Related Work

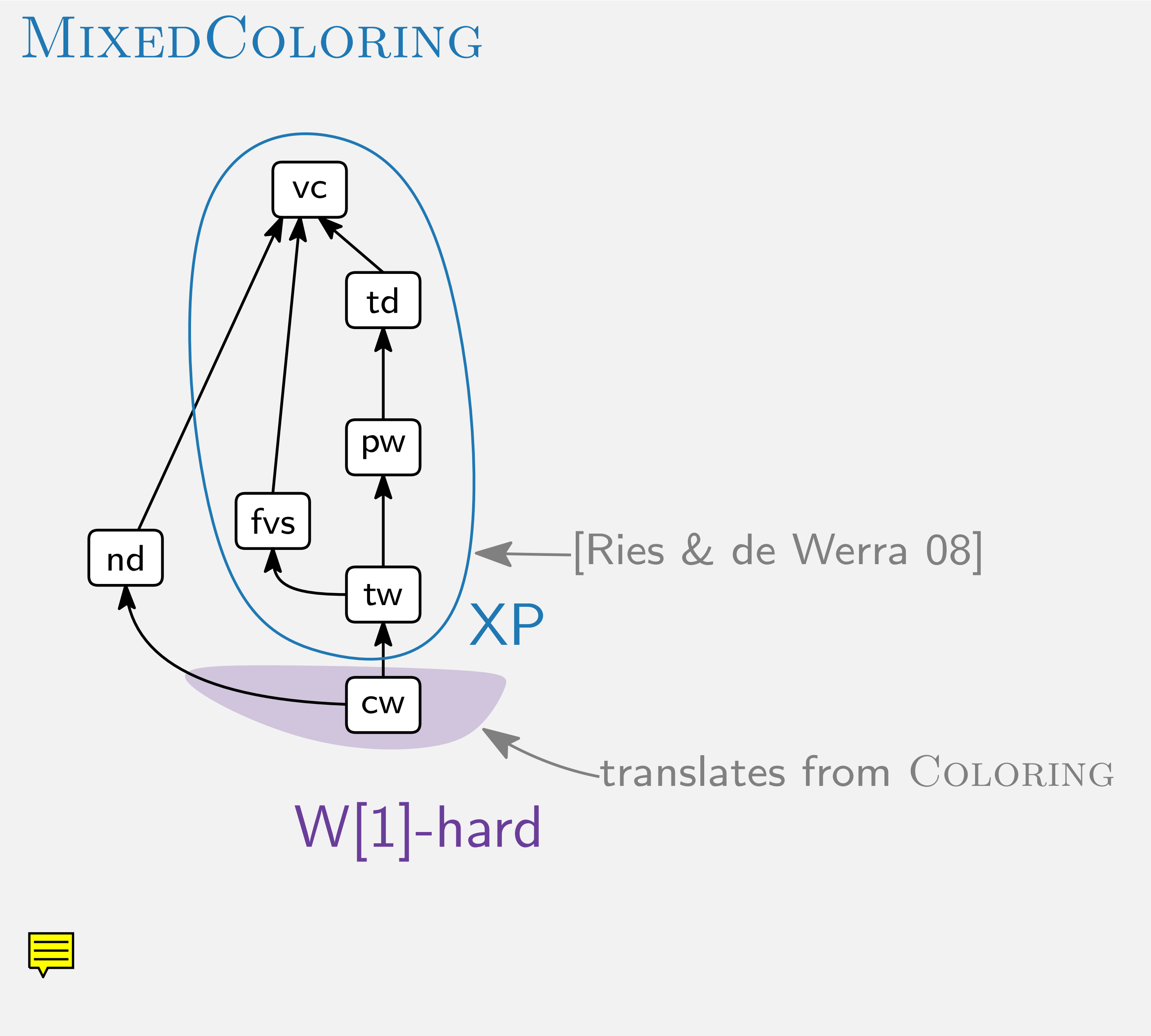
MIXEDCOLORING

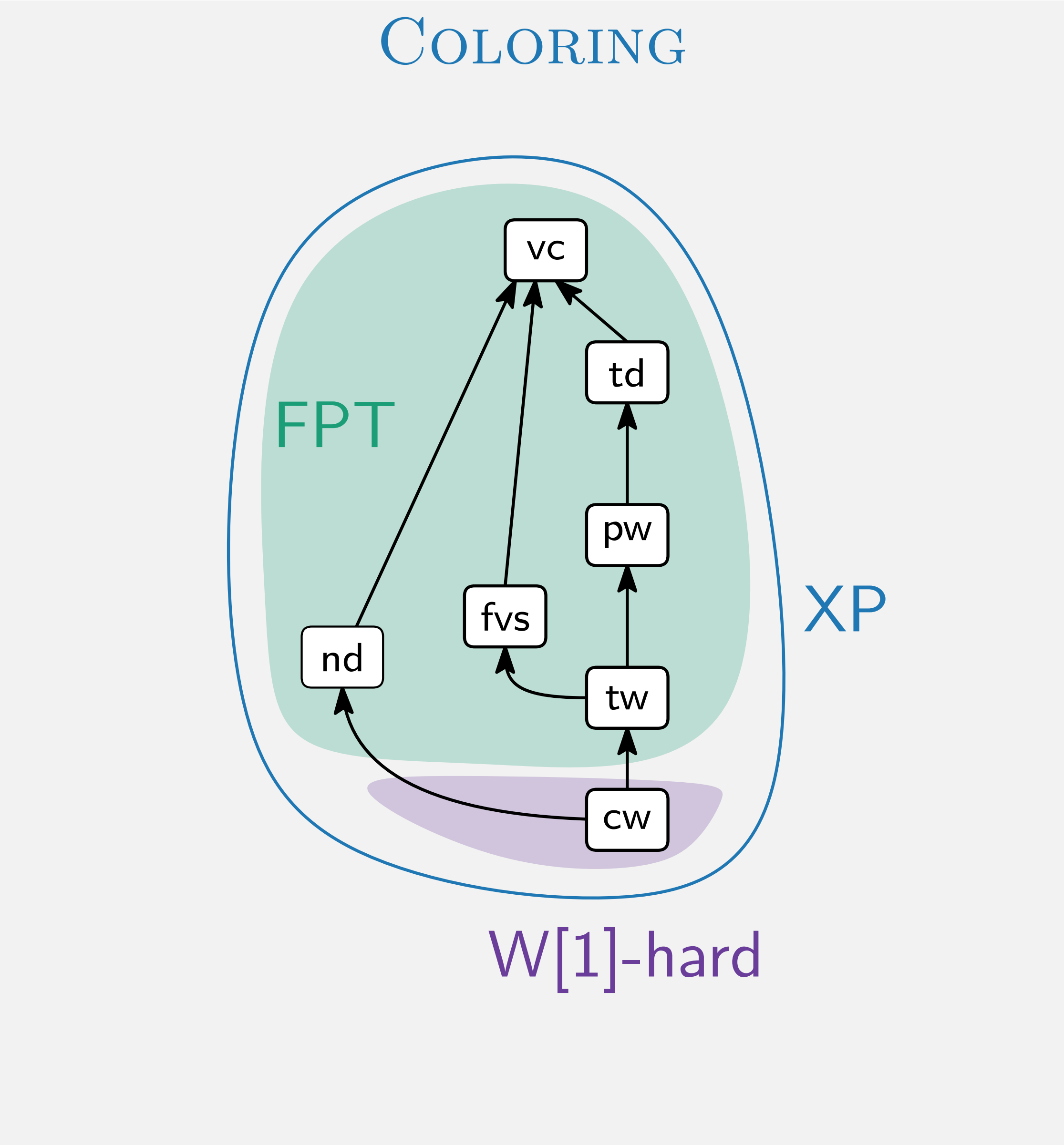


COLORING

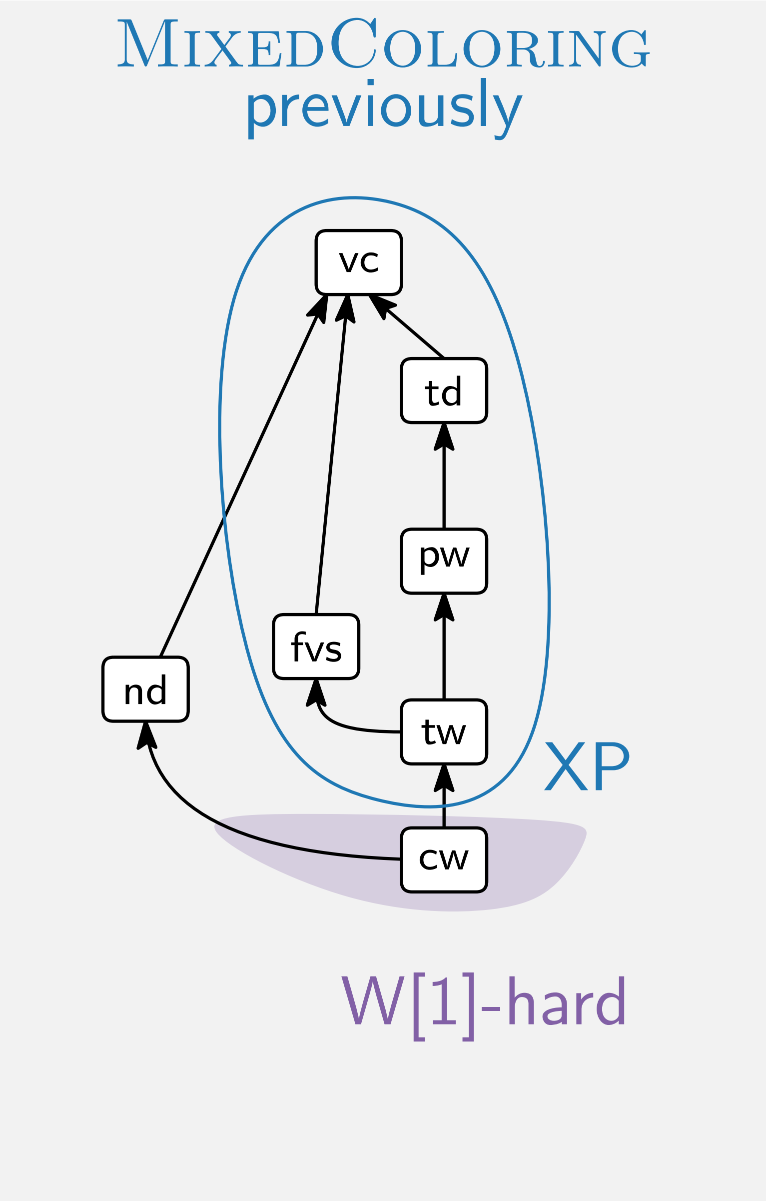


Related Work



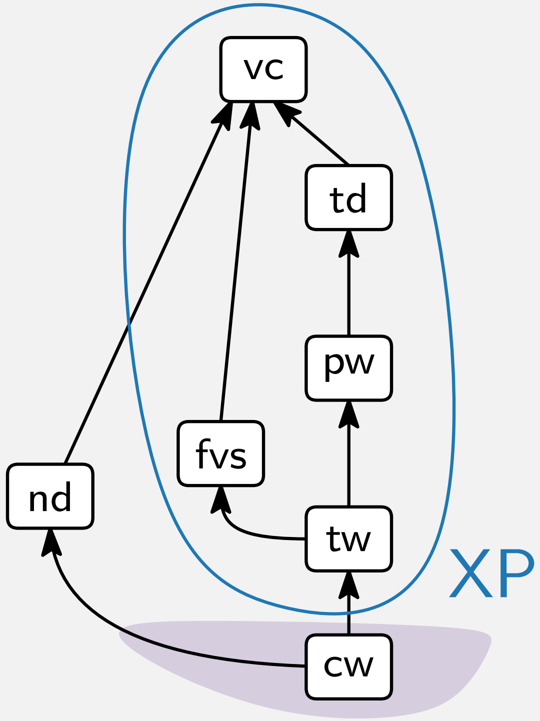


Parameterized Complexity



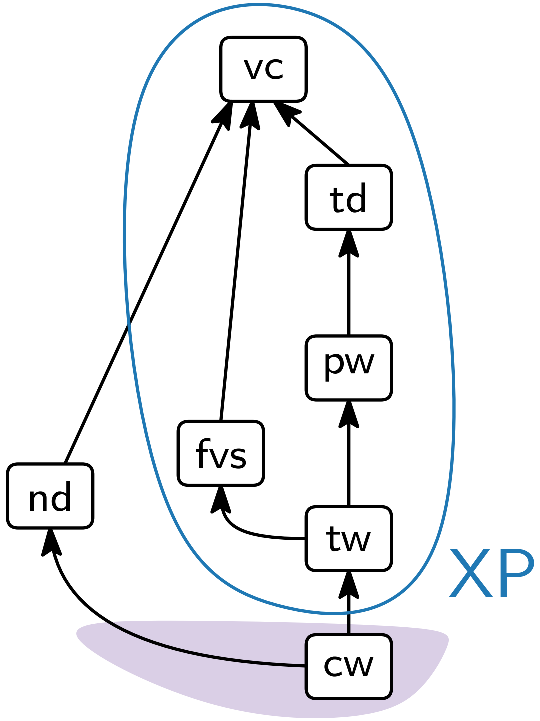
Parameterized Complexity

MixedColoring
previously



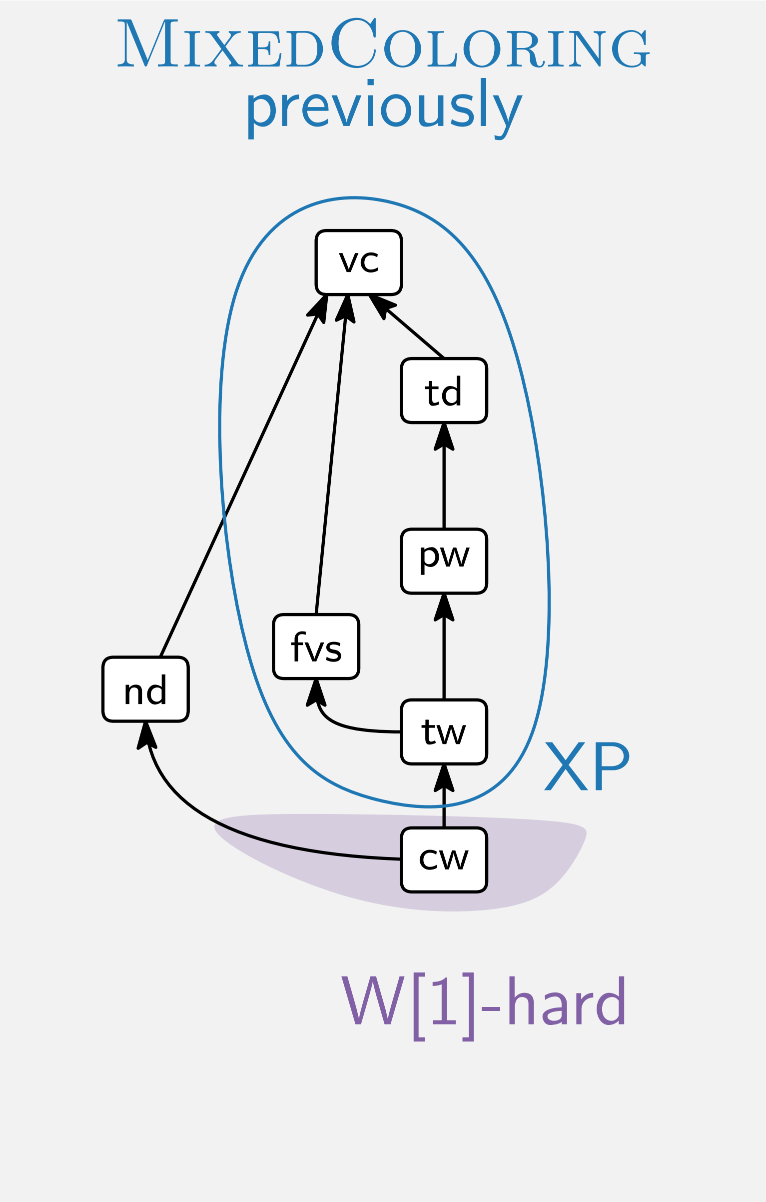
W[1]-hard

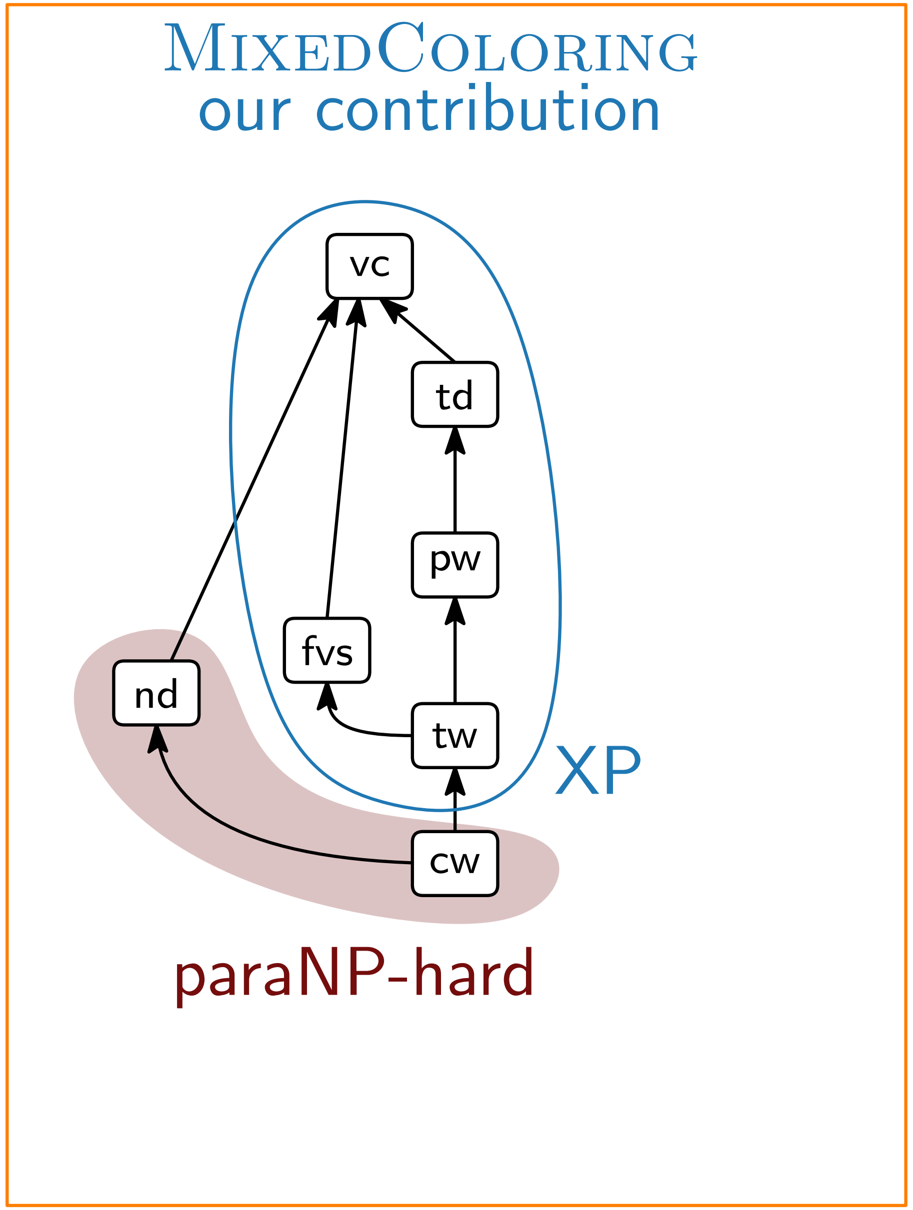
MixedColoring
our contribution



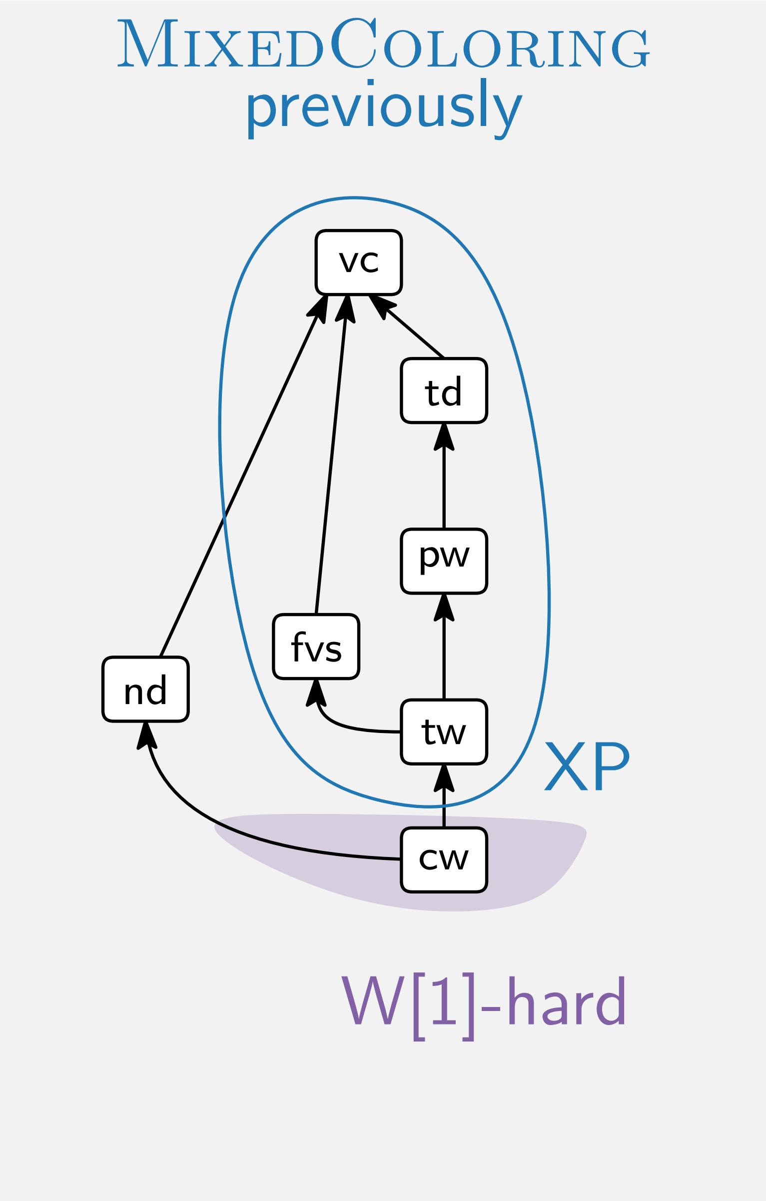
W[1]-hard

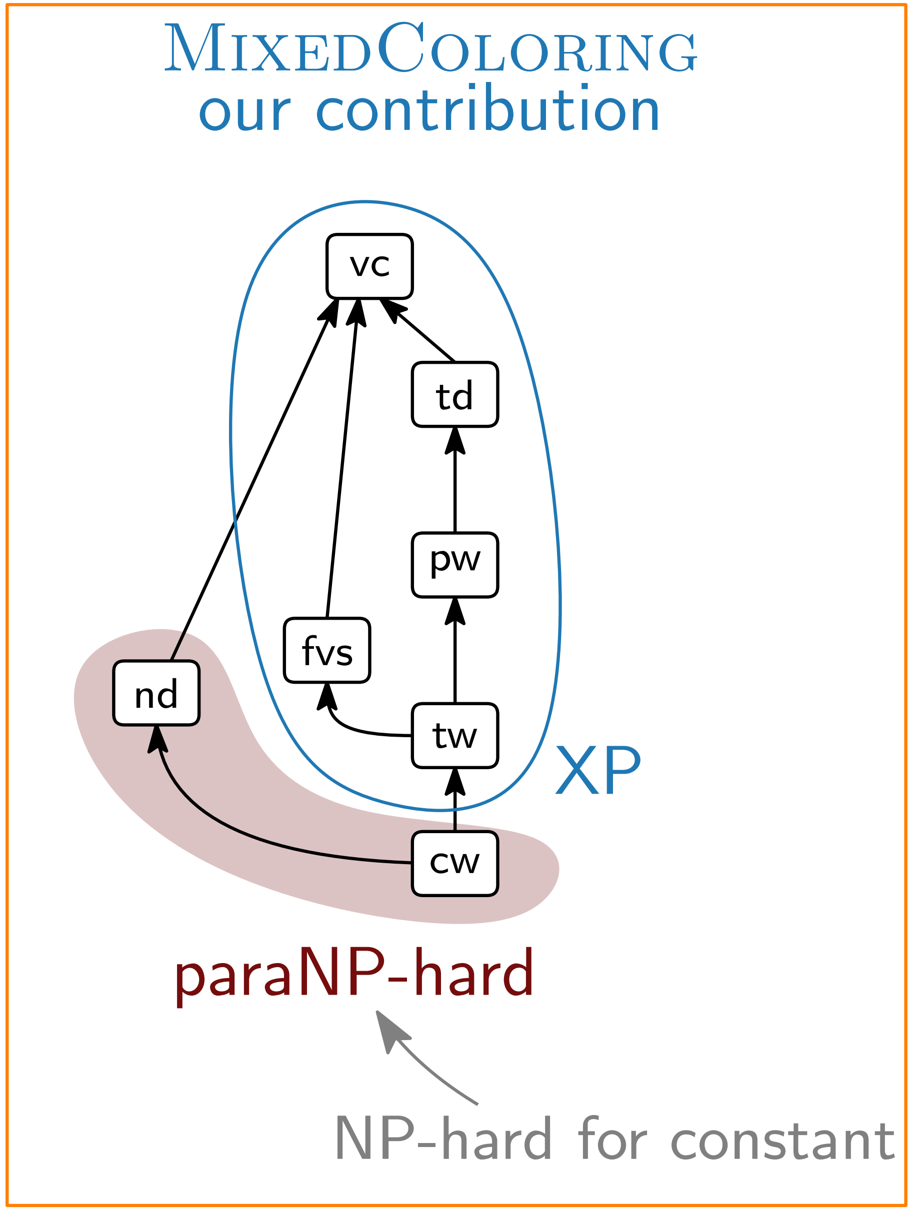
Parameterized Complexity


MixedColoring
previously


MixedColoring
our contribution

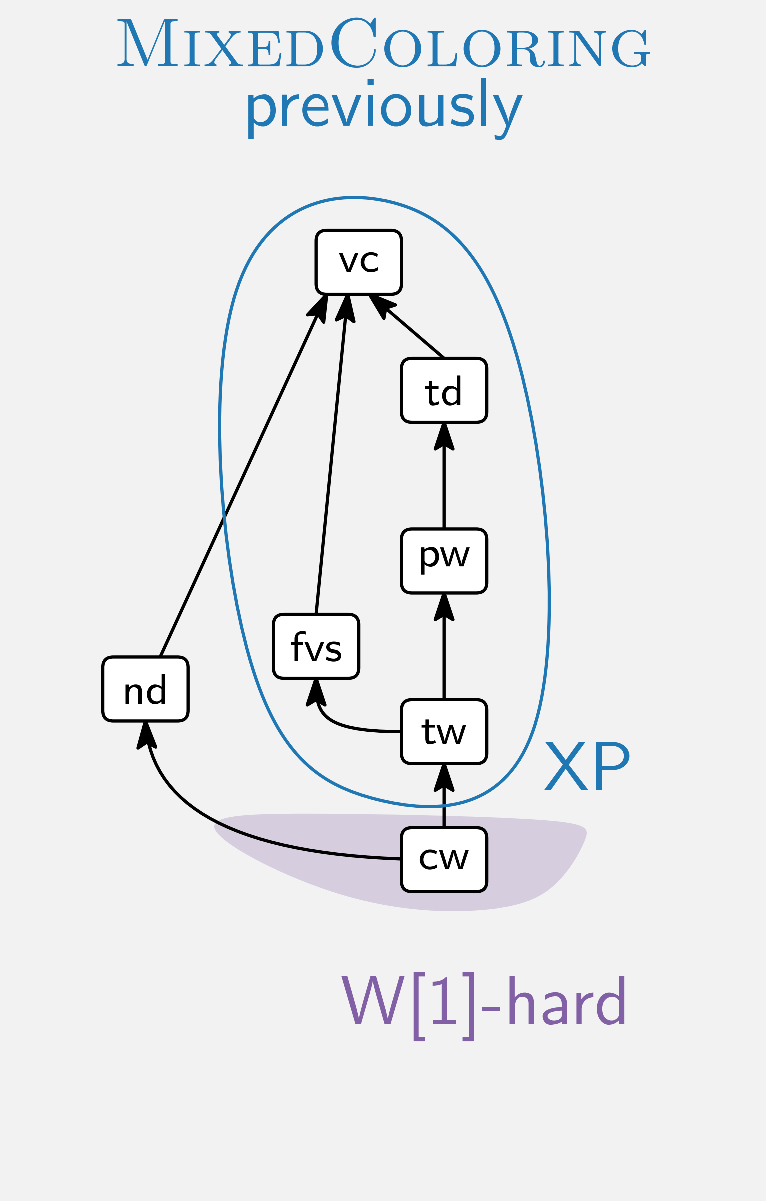
Parameterized Complexity

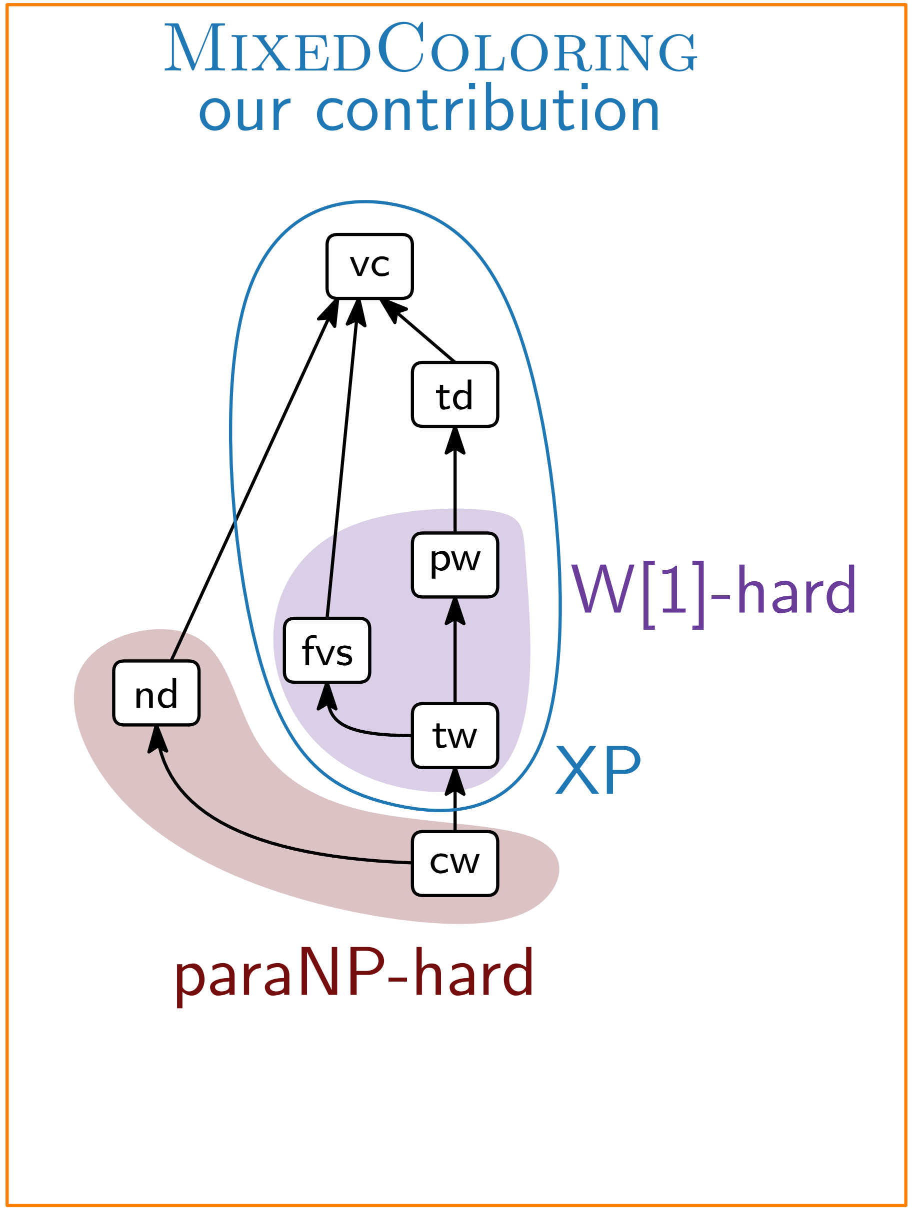




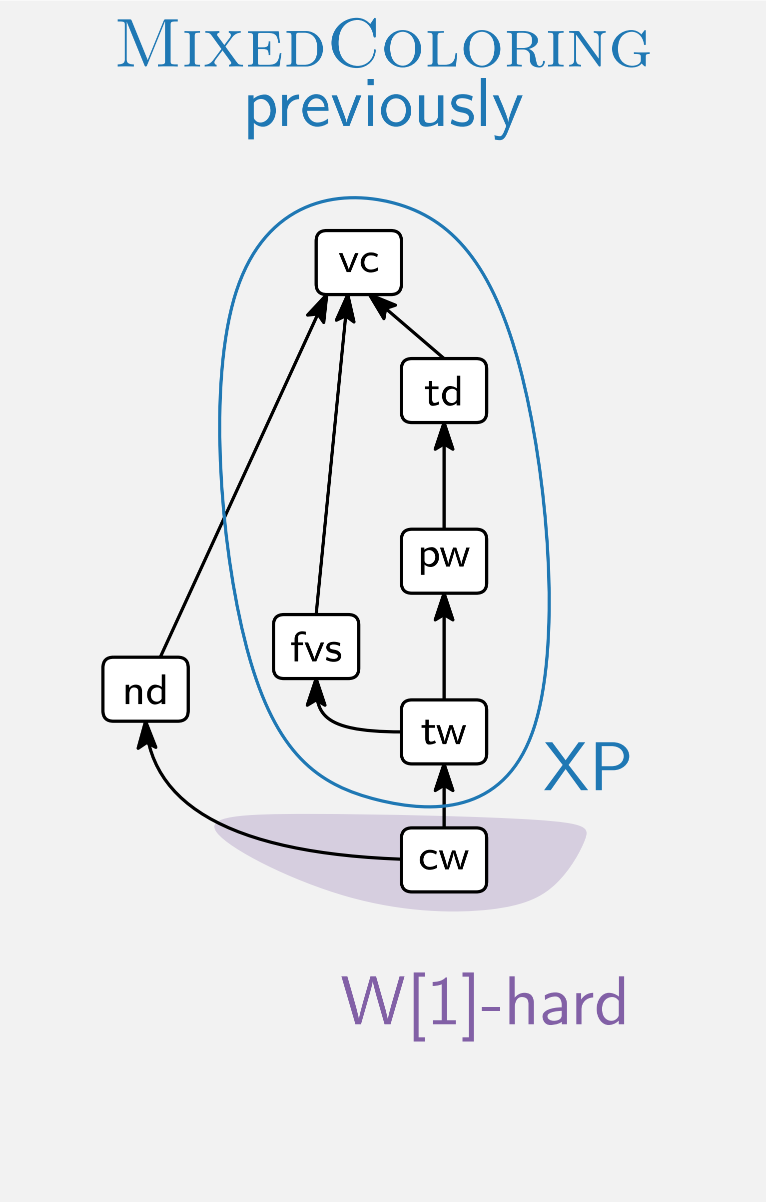
NP-hard for constant values of parameter

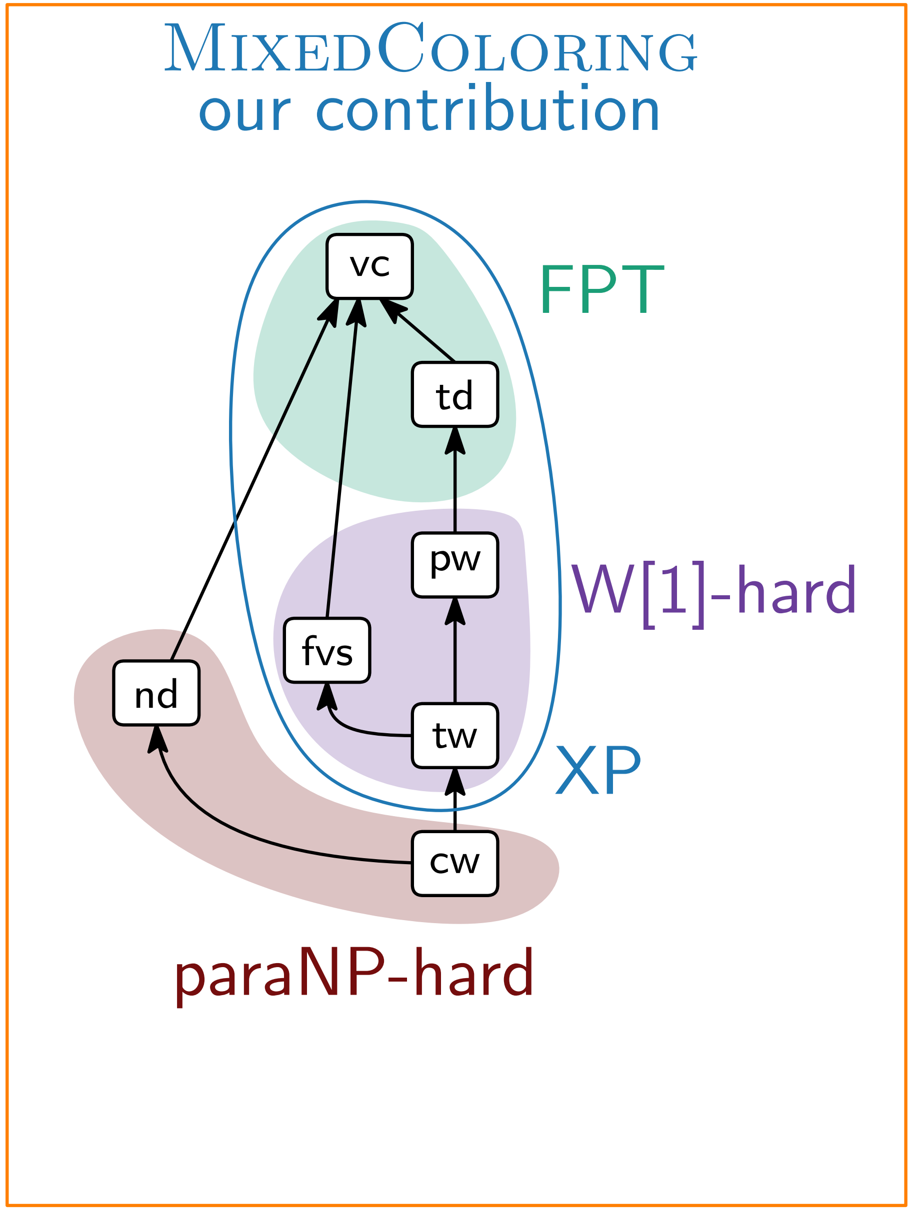
Parameterized Complexity





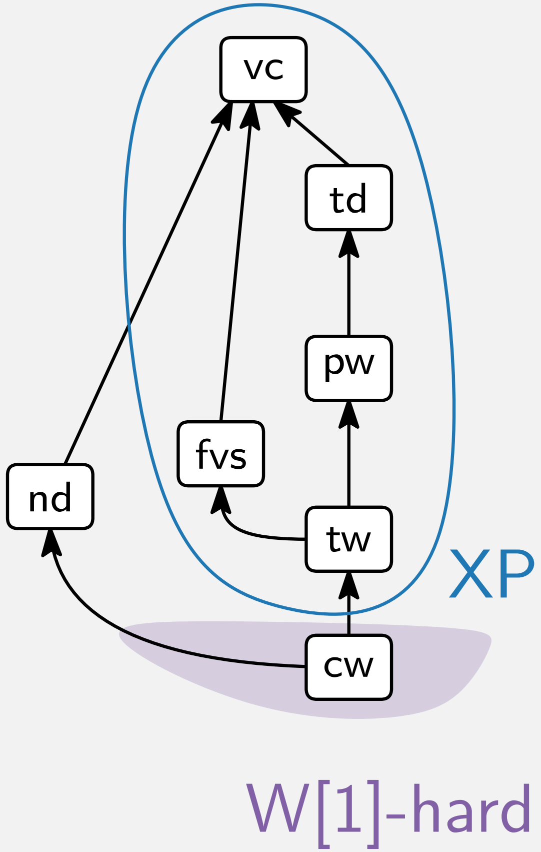
Parameterized Complexity



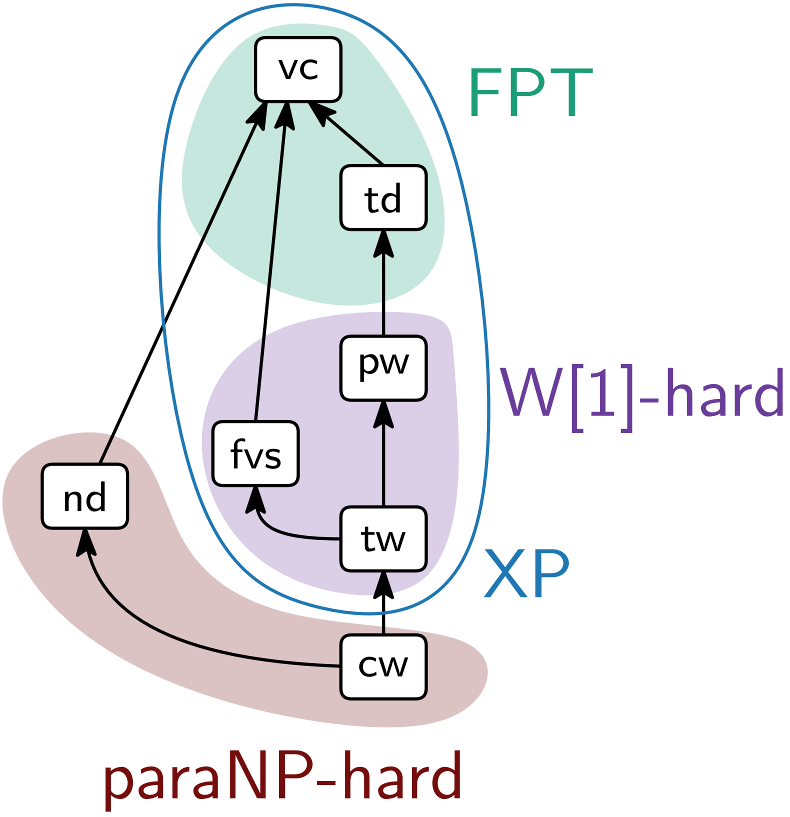


Parameterized Complexity

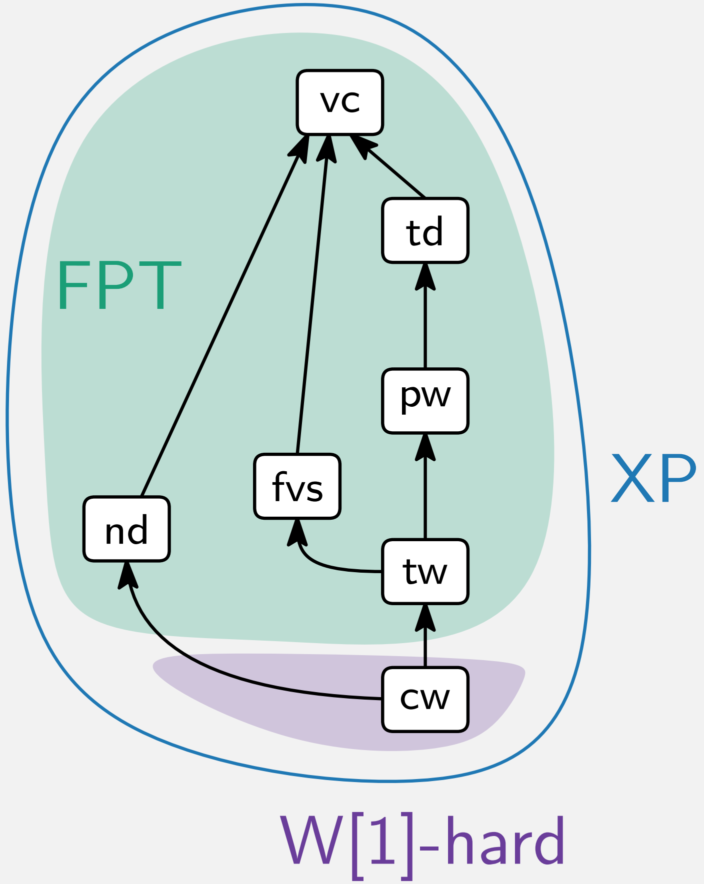
MixedColoring
previously



MixedColoring
our contribution



Coloring



Mixed Parameters

Neighborhood Diversity

Mixed Parameters

Neighborhood Diversity

- counts types of vertices

Mixed Parameters

Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$

Mixed Parameters

Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff
  $N(u) \setminus \{v\} = N(v) \setminus \{u\}$

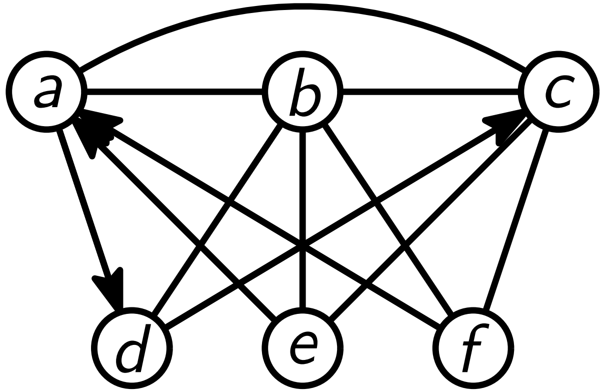
  neighborhood

Mixed Parameters

Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$

neighborhood

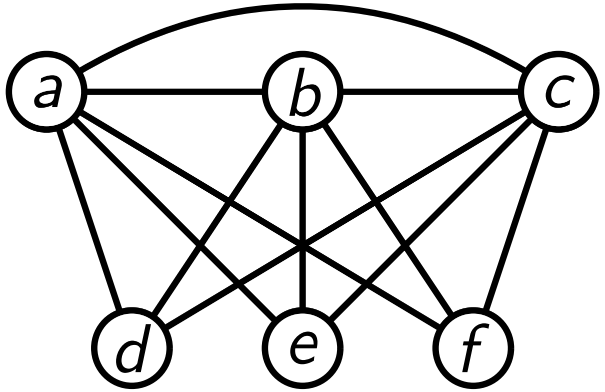


Mixed Parameters

Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$

neighborhood

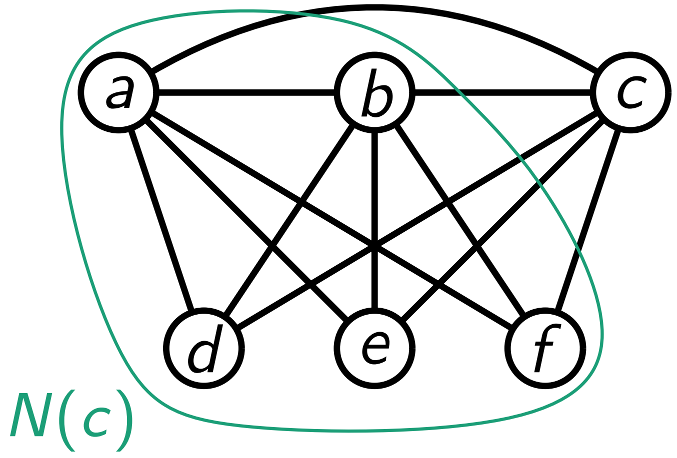


Mixed Parameters

Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$

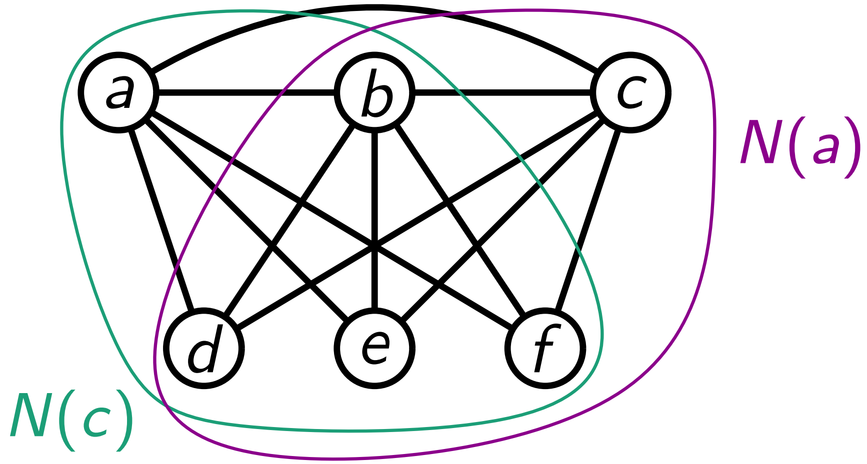
neighborhood



Mixed Parameters

Neighborhood Diversity

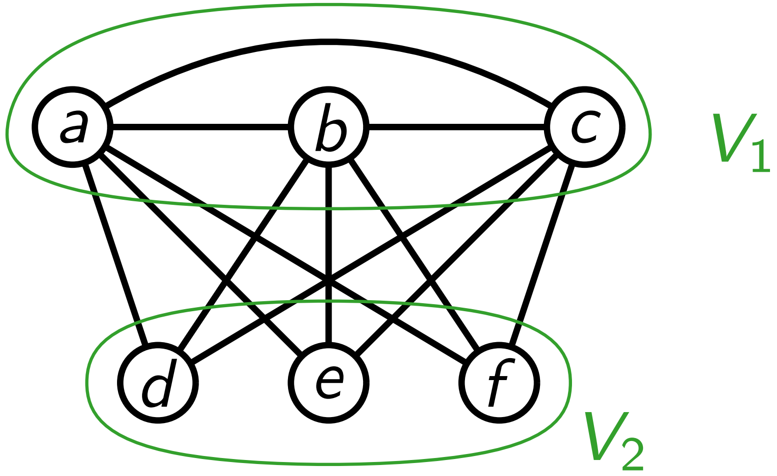
- counts types of vertices
- vertices u, v of same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$



Mixed Parameters

Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$

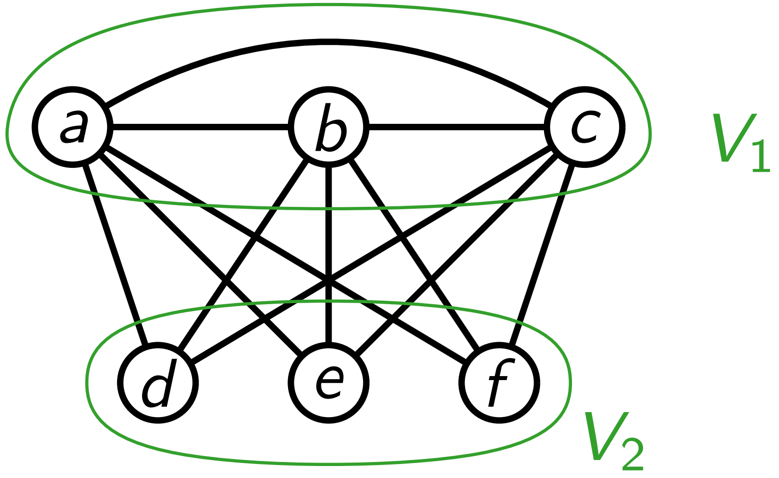


Mixed Parameters

Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$

Observation



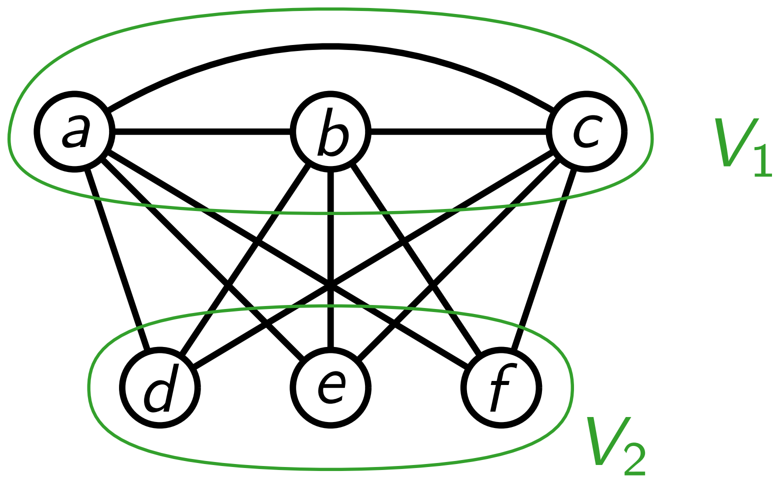
Mixed Parameters

Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$

Observation

- subgraphs induced by types form clique or independent set



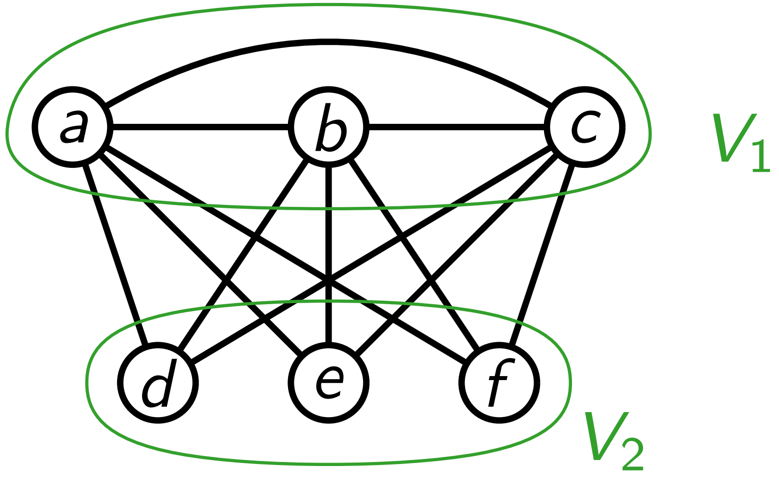
Mixed Parameters

Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$

Observation

- subgraphs induced by types form clique or independent set
- vertices in type inducing an independent set can receive same color



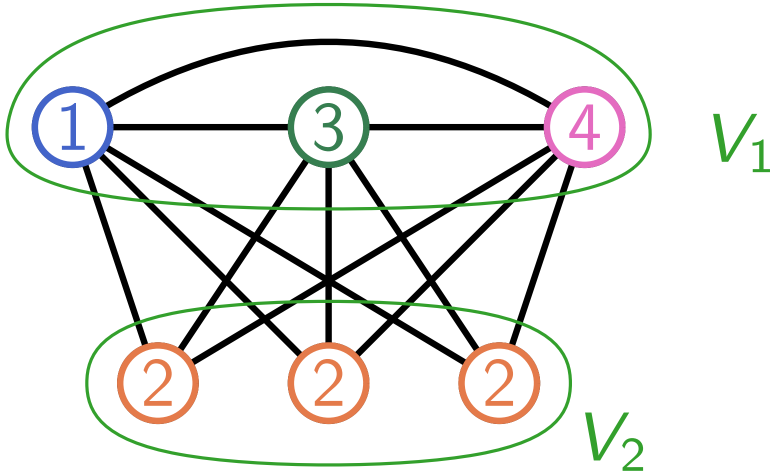
Mixed Parameters

Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$

Observation

- subgraphs induced by types form clique or independent set
- vertices in type inducing an independent set can receive same color



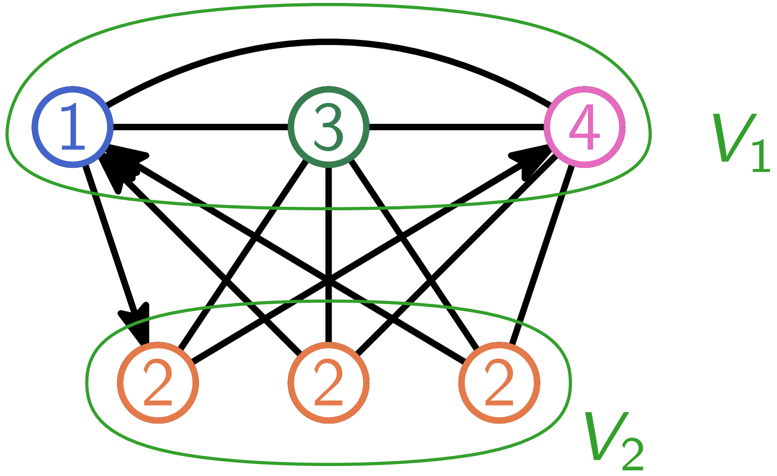
Mixed Parameters

Neighborhood Diversity

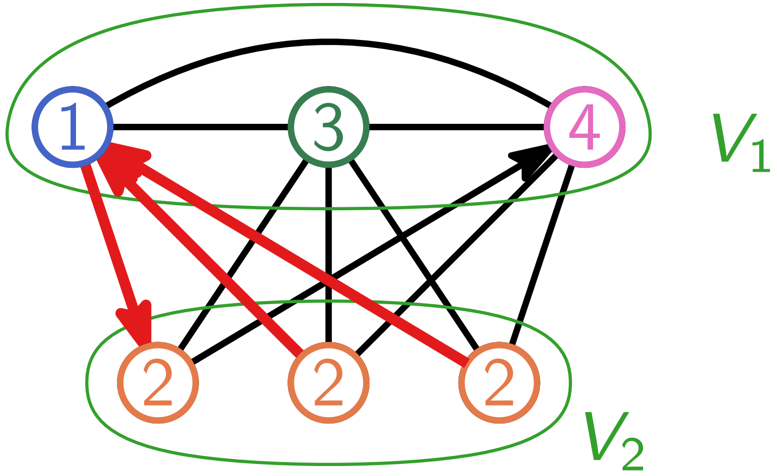
- counts types of vertices
- vertices u, v of same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$

Observation

- subgraphs induced by types form clique or independent set
- vertices in type inducing an independent set can receive same color



Mixed Parameters



Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$

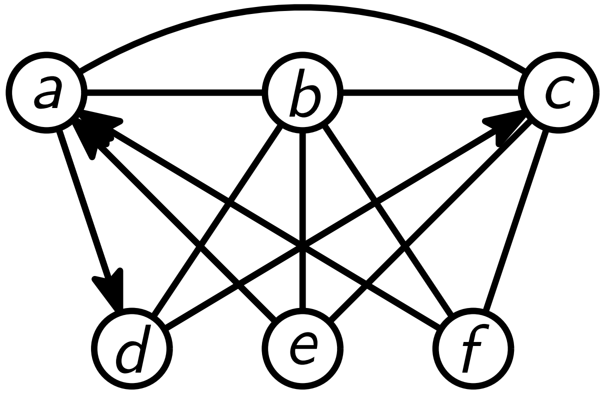
Observation

- subgraphs induced by types form clique or independent set
- vertices in type inducing an independent set can receive same color

doesn't work with arcs

Mixed Parameters

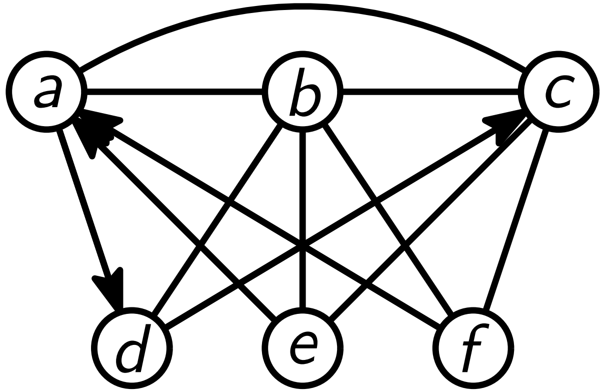
Mixed Neighborhood Diversity



Mixed Parameters

Mixed Neighborhood Diversity

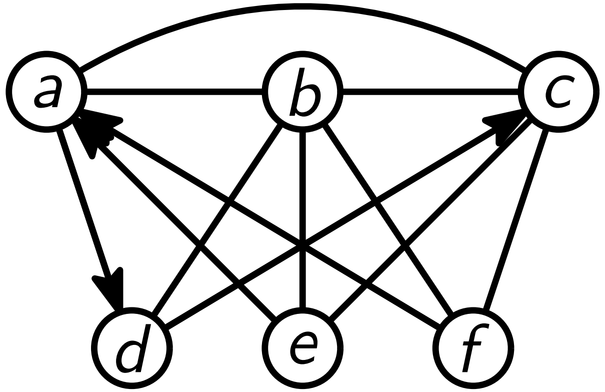
- counts types of vertices



Mixed Parameters

Mixed Neighborhood Diversity

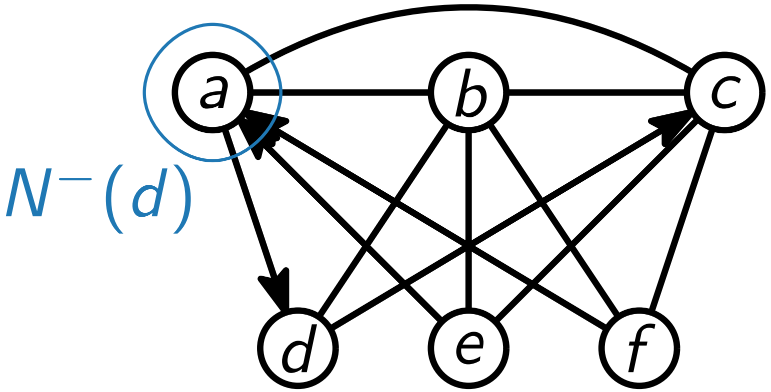
- counts types of vertices
- vertices u, v of same type iff



Mixed Parameters

Mixed Neighborhood Diversity

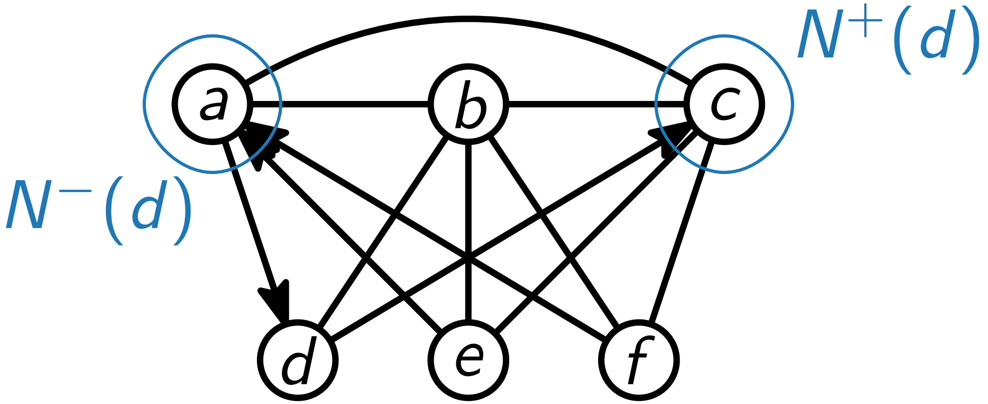
- counts types of vertices
- vertices u, v of same type iff
  - $N^{-}(u) = N^{-}(v)$



Mixed Parameters

Mixed Neighborhood Diversity

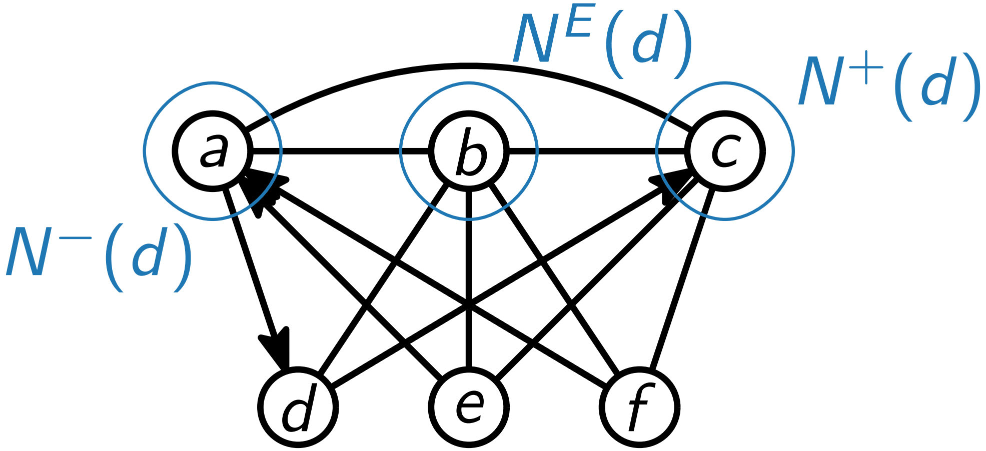
- counts types of vertices
- vertices u, v of same type iff
  - $N^-(u) = N^-(v)$
  - $N^+(u) = N^+(v)$



Mixed Parameters

Mixed Neighborhood Diversity

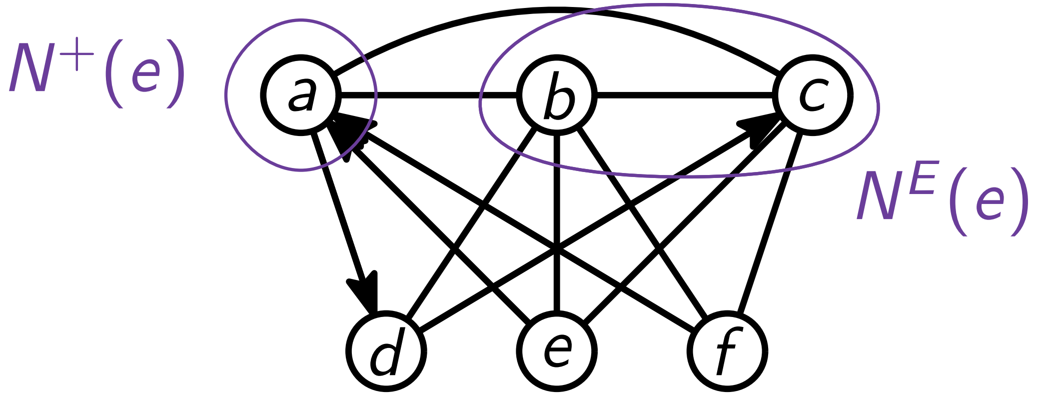
- counts types of vertices
- vertices u, v of same type iff
  - $N^-(u) = N^-(v)$
  - $N^+(u) = N^+(v)$
  - $N^E(u) \setminus \{v\} = N^E(v) \setminus \{u\}$



Mixed Parameters

Mixed Neighborhood Diversity

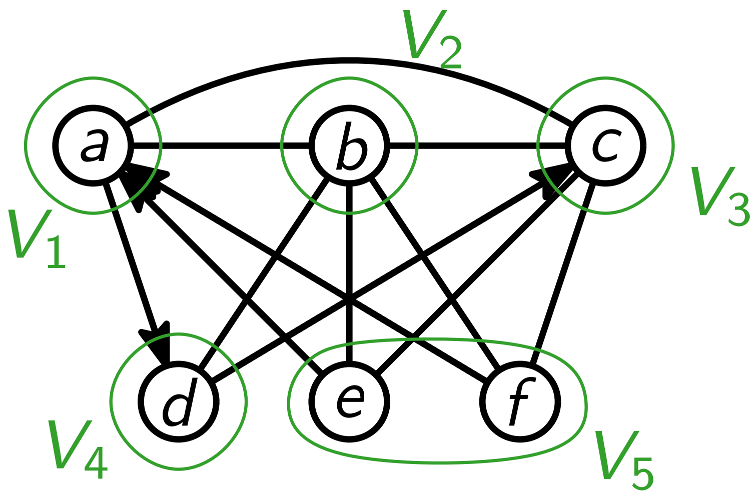
- counts types of vertices
- vertices u, v of same type iff
    - $N^-(u) = N^-(v)$
    - $N^+(u) = N^+(v)$
    - $N^E(u) \setminus \{v\} = N^E(v) \setminus \{u\}$



Mixed Parameters

Mixed Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff
  - $N^-(u) = N^-(v)$
  - $N^+(u) = N^+(v)$
  - $N^E(u) \setminus \{v\} = N^E(v) \setminus \{u\}$



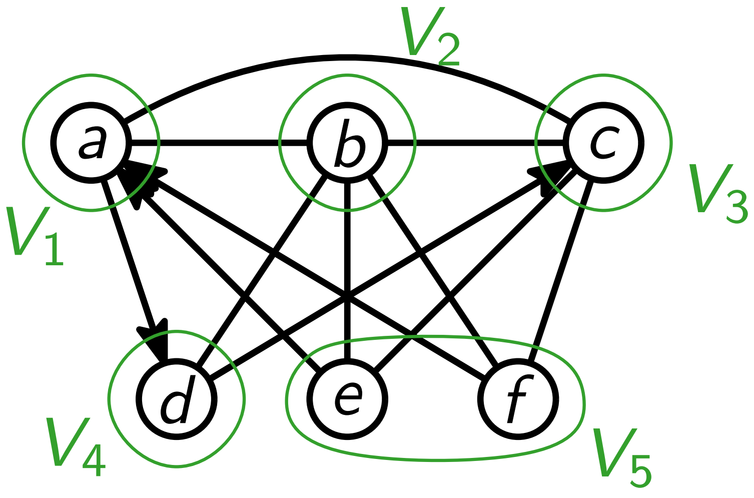
Mixed Parameters

Mixed Neighborhood Diversity

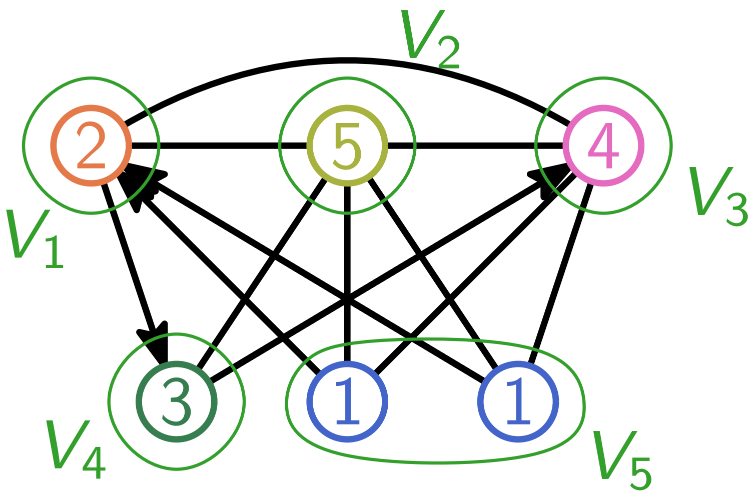
- counts types of vertices
- vertices u, v of same type iff
  - $N^-(u) = N^-(v)$
  - $N^+(u) = N^+(v)$
  - $N^E(u) \setminus \{v\} = N^E(v) \setminus \{u\}$

Observation

- subgraphs induced by types form clique or independent set
- vertices in type inducing IS can receive same color



Mixed Parameters



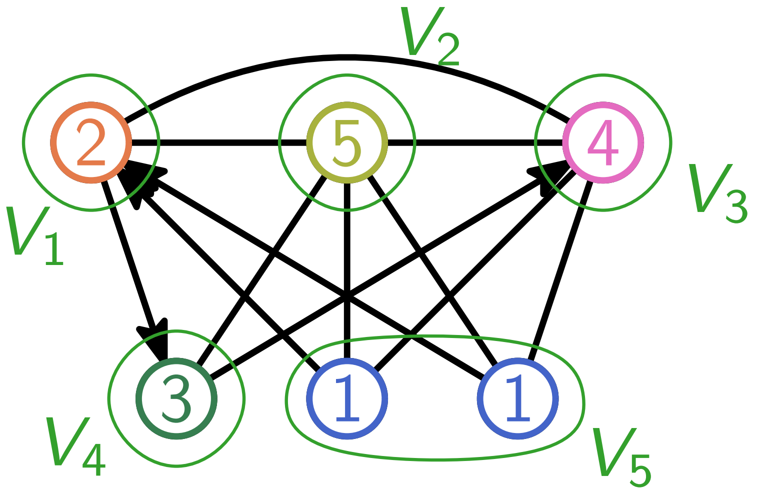
Mixed Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff
  - $N^-(u) = N^-(v)$
  - $N^+(u) = N^+(v)$
  - $N^E(u) \setminus \{v\} = N^E(v) \setminus \{u\}$

Observation

- subgraphs induced by types form clique or independent set
- vertices in type inducing IS can receive same color

Mixed Parameters



Mixed Neighborhood Diversity

- counts types of vertices
- vertices u, v of same type iff
  - $N^-(u) = N^-(v)$
  - $N^+(u) = N^+(v)$
  - $N^E(u) \setminus \{v\} = N^E(v) \setminus \{u\}$

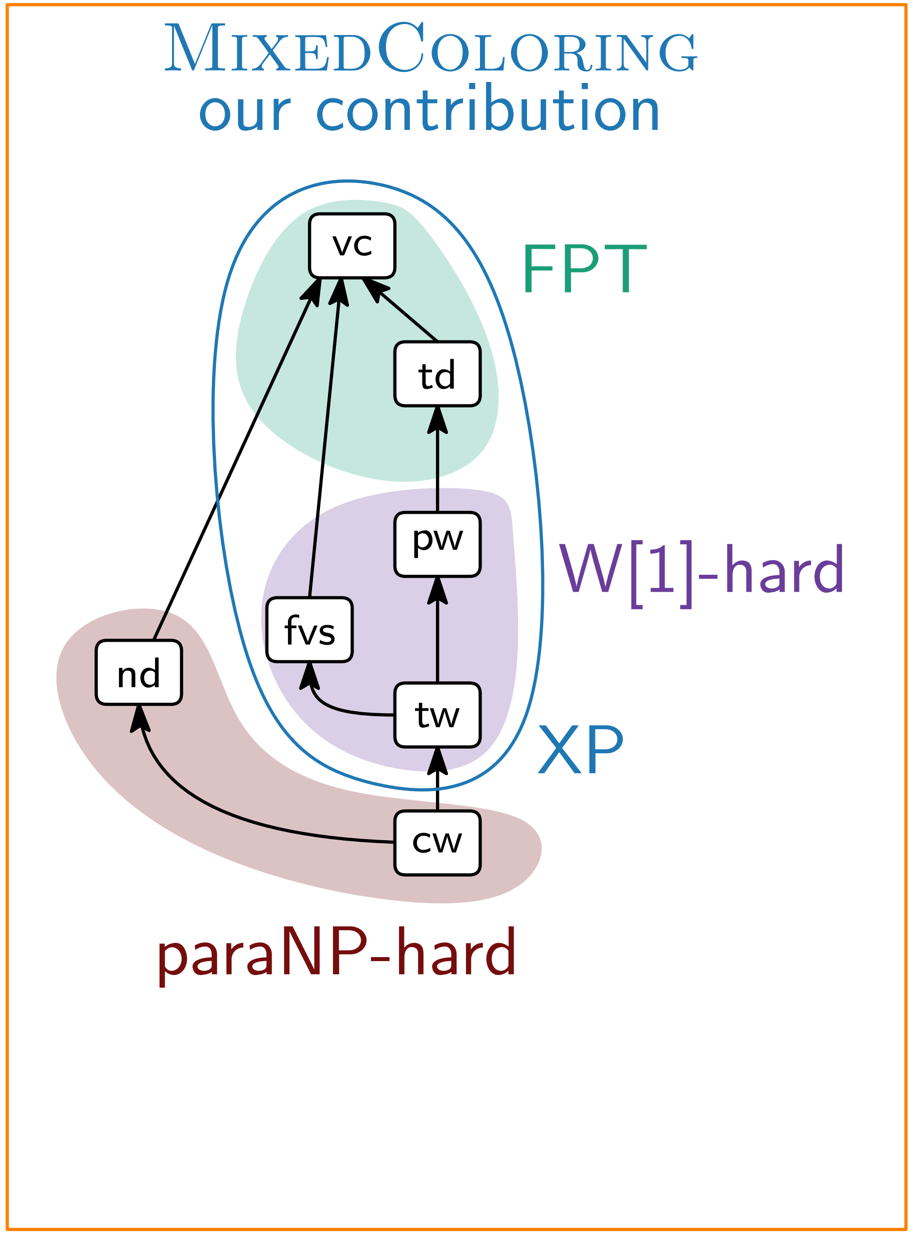
Observation

- subgraphs induced by types form clique or independent set
- vertices in type inducing IS can receive same color

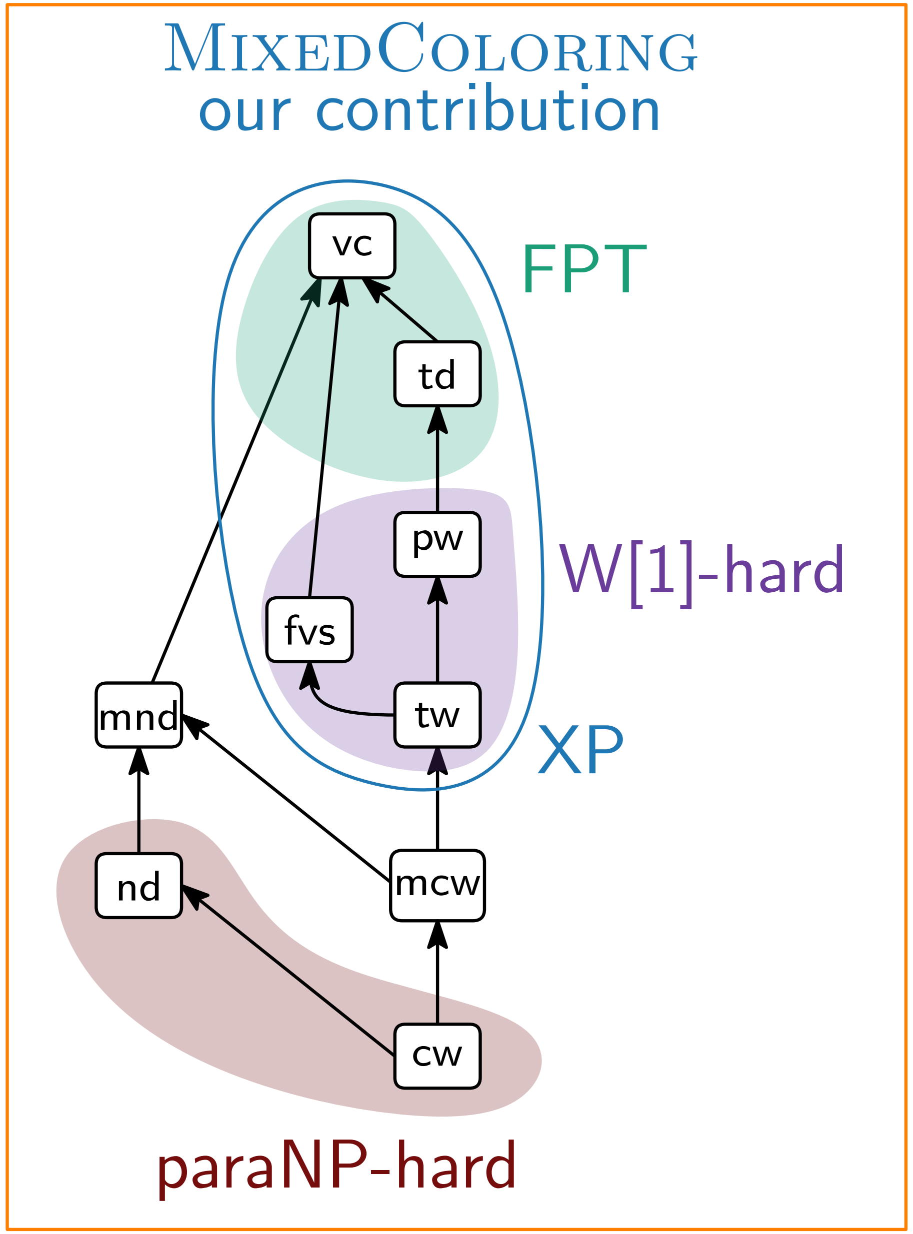
Mixed Cliquewidth

combines definitions of cliquewidth for undirected and directed graphs

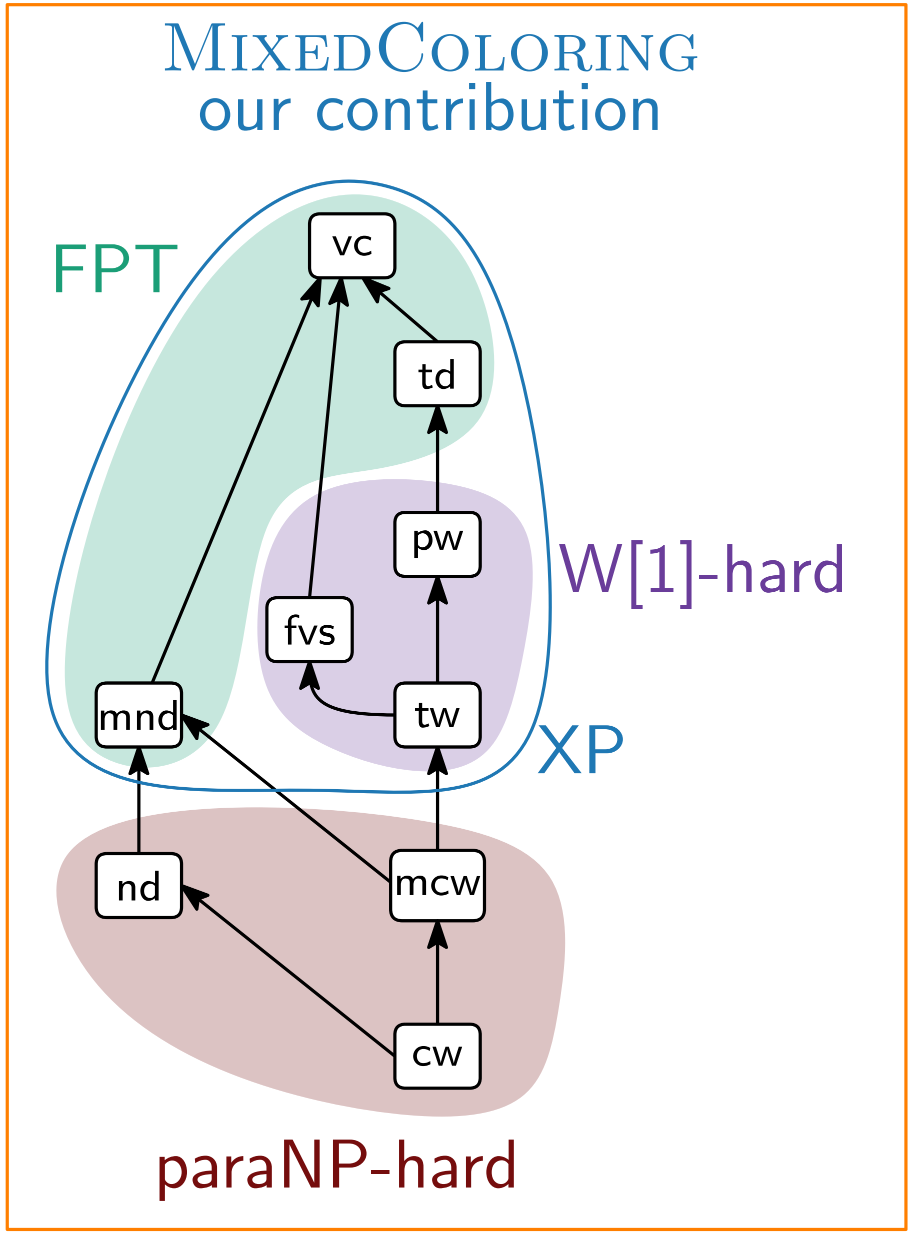
Parameterized Complexity



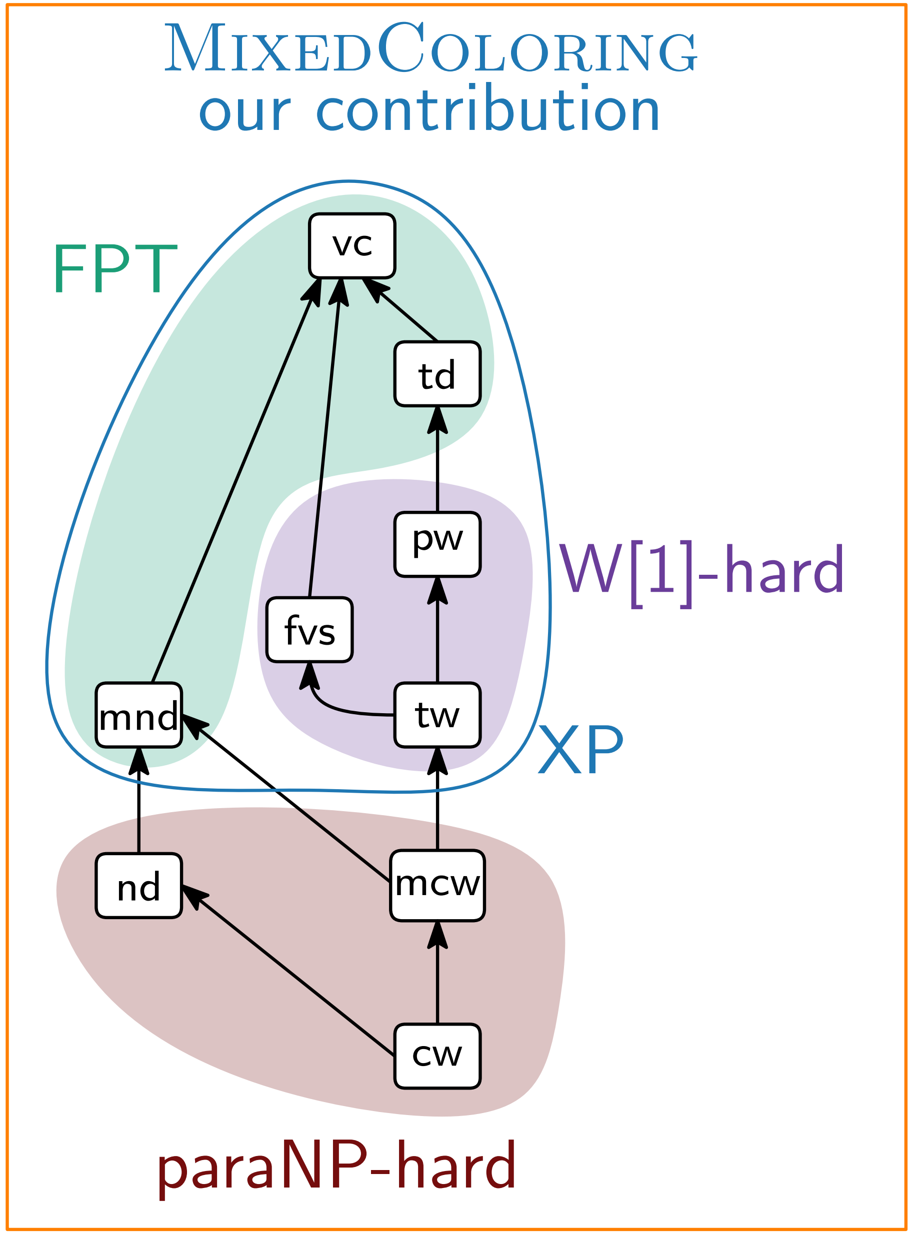
Parameterized Complexity

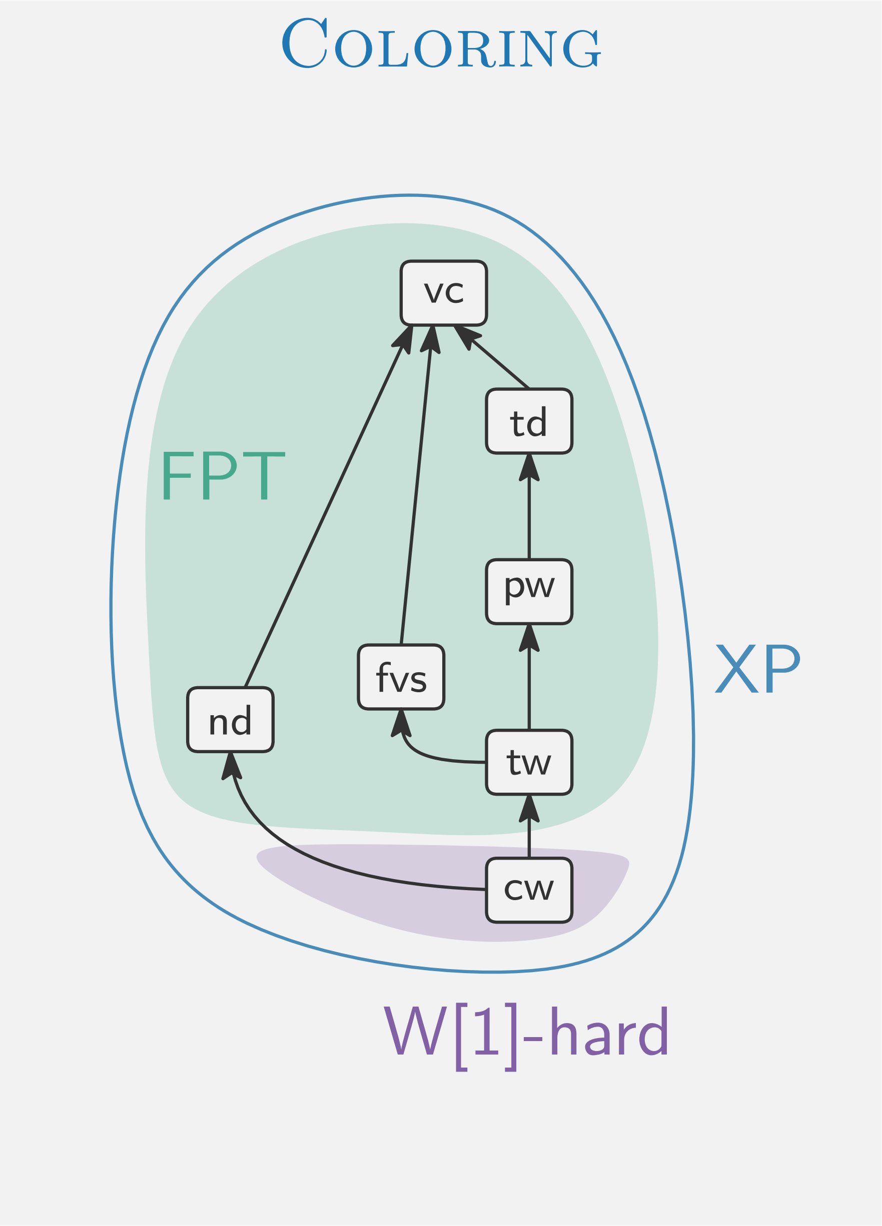


Parameterized Complexity

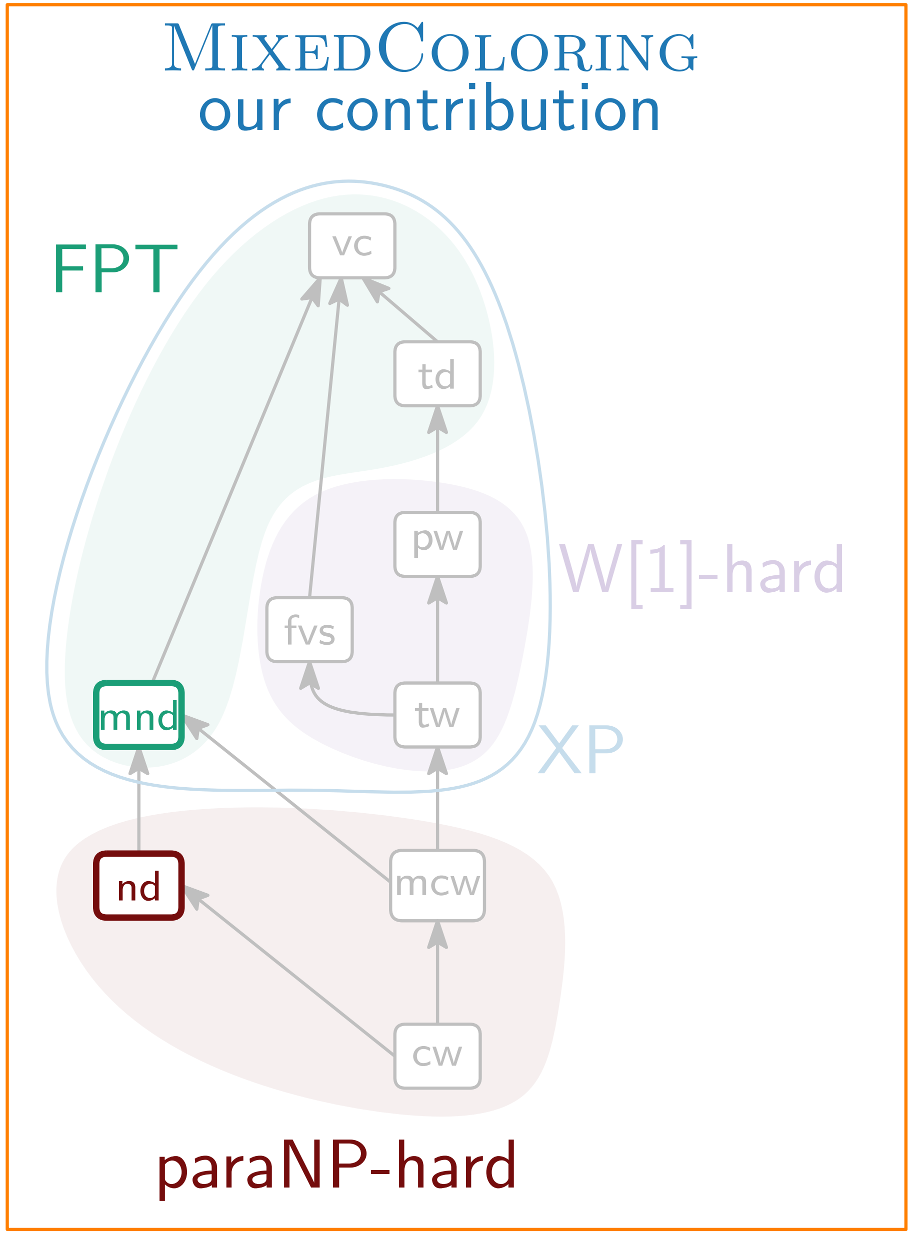


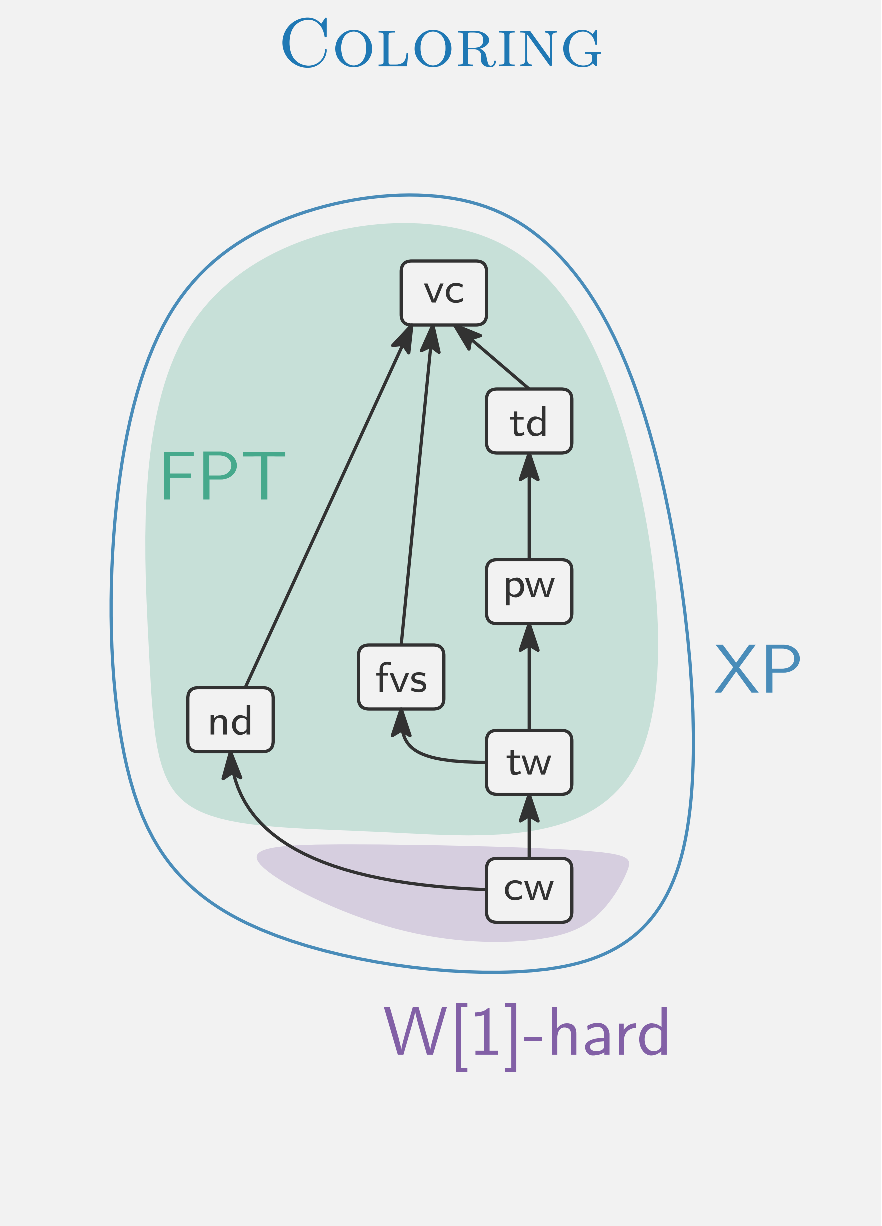
Parameterized Complexity





Parameterized Complexity





paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from
PRECEDENCE CONSTRAINED SCHEDULING

NP-hard [Garey & Johnson, SS9]

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input:

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input: ◆ sets $T_{\hexagon}$, $T_{\pentagon}$ of unit-length tasks

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input: ♦ sets $T_{\hexagon}$, $T_{\pentagon}$ of unit-length tasks

$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input:
- sets $T_{\hexagon}$, $T_{\pentagon}$ of unit-length tasks
- partial order $\prec$ of tasks

$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input:
- sets $T_{\hexagon}$, $T_{\pentagon}$ of unit-length tasks
- partial order $\prec$ of tasks

$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input:
- sets $T_{\hexagon}, T_{\pentagon}$ of unit-length tasks
- partial order $\prec$ of tasks
- deadline D

$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

Precedence Constrained Scheduling

Input:
- sets $T_{⬡}$, $T_{⬠}$ of unit-length tasks
- partial order $\prec$ of tasks
- deadline D

$T_{⬡} = \{t_1, t_5\}$

$T_{⬠} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

Precedence Constrained Scheduling

Input:
- sets $T_{\hexagon}$, $T_{\pentagon}$ of unit-length tasks
- partial order $\prec$ of tasks
- deadline D

Output:

$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input:
- sets $T_{\hexagon}$, $T_{\pentagon}$ of unit-length tasks
- partial order $\prec$ of tasks
- deadline D

Output: schedule on two machines $M_{\hexagon}$, $M_{\pentagon}$
respecting partial order $\prec$ and deadline D
$(T_{\hexagon}$ on $M_{\hexagon}$, $T_{\pentagon}$ on $M_{\pentagon})$

$T_{\hexagon} = \{t_1, t_5\}$
$T_{\pentagon} = \{t_2, t_3, t_4\}$
$t_1 \prec t_3 \prec t_4$
$t_2 \prec t_5$
$D = 3$

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input:
- sets $T_{\hexagon}$, $T_{\pentagon}$ of unit-length tasks
- partial order $\prec$ of tasks
- deadline D

Output: schedule on two machines $M_{\hexagon}$, $M_{\pentagon}$
respecting partial order $\prec$ and deadline D
($T_{\hexagon}$ on $M_{\hexagon}$, $T_{\pentagon}$ on $M_{\pentagon}$)

$T_{\hexagon} = \{t_1, t_5\}$
$T_{\pentagon} = \{t_2, t_3, t_4\}$
$t_1 \prec t_3 \prec t_4$
$t_2 \prec t_5$
$D = 3$

Reduction (Idea):

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input:
- sets $T_{\hexagon}$, $T_{\pentagon}$ of unit-length tasks
- partial order $\prec$ of tasks
- deadline D

Output: schedule on two machines $M_{\hexagon}$, $M_{\pentagon}$
respecting partial order $\prec$ and deadline D
$(T_{\hexagon}$ on $M_{\hexagon}$, $T_{\pentagon}$ on $M_{\pentagon})$

$T_{\hexagon} = \{t_1, t_5\}$
$T_{\pentagon} = \{t_2, t_3, t_4\}$
$t_1 \prec t_3 \prec t_4$
$t_2 \prec t_5$
$D = 3$

Reduction (Idea):

- tasks as vertices

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input:
- sets $T_{\hexagon}$, $T_{\pentagon}$ of unit-length tasks
- partial order $\prec$ of tasks
- deadline D

Output: schedule on two machines $M_{\hexagon}$, $M_{\pentagon}$
respecting partial order $\prec$ and deadline D
($T_{\hexagon}$ on $M_{\hexagon}$, $T_{\pentagon}$ on $M_{\pentagon}$)

$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

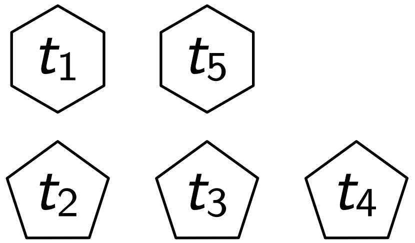
$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$

Reduction (Idea):

- tasks as vertices



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input:
- ♦ sets $T_{\hexagon}$, $T_{\pentagon}$ of unit-length tasks
- ♦ partial order $\prec$ of tasks
- ♦ deadline D

Output: schedule on two machines $M_{\hexagon}$, $M_{\pentagon}$ respecting partial order $\prec$ and deadline D ($T_{\hexagon}$ on $M_{\hexagon}$, $T_{\pentagon}$ on $M_{\pentagon}$)

$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

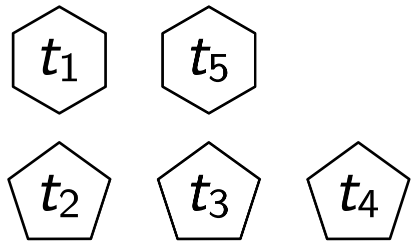
$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$

Reduction (Idea):

- tasks as vertices
- timeslots as colors



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input:
- ♦ sets $T_{\hexagon}$, $T_{\pentagon}$ of unit-length tasks
- ♦ partial order $\prec$ of tasks
- ♦ deadline D

Output: schedule on two machines $M_{\hexagon}$, $M_{\pentagon}$
respecting partial order $\prec$ and deadline D
($T_{\hexagon}$ on $M_{\hexagon}$, $T_{\pentagon}$ on $M_{\pentagon}$)

$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

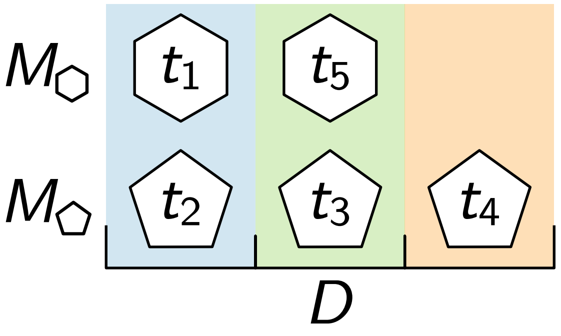
$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$

Reduction (Idea):

- tasks as vertices
- timeslots as colors



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

PRECEDENCE CONSTRAINED SCHEDULING

Input:
- sets $T_{\text{⬡}}$, $T_{\text{⬠}}$ of unit-length tasks
- partial order $\prec$ of tasks
- deadline D

Output: schedule on two machines $M_{\text{⬡}}$, $M_{\text{⬠}}$
respecting partial order $\prec$ and deadline D
($T_{\text{⬡}}$ on $M_{\text{⬡}}$, $T_{\text{⬠}}$ on $M_{\text{⬠}}$)

$T_{\text{⬡}} = \{t_1, t_5\}$

$T_{\text{⬠}} = \{t_2, t_3, t_4\}$

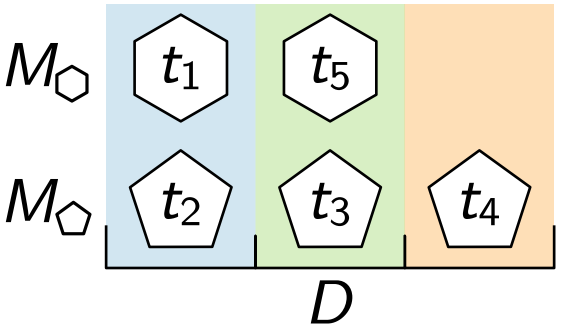
$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$

Reduction (Idea):

- tasks as vertices
- timeslots as colors
- arcs enforce partial order



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction from

Precedence Constrained Scheduling

Input:
- sets $T_{\hexagon}$, $T_{\pentagon}$ of unit-length tasks
- partial order $\prec$ of tasks
- deadline D

Output: schedule on two machines $M_{\hexagon}$, $M_{\pentagon}$ respecting partial order $\prec$ and deadline D ($T_{\hexagon}$ on $M_{\hexagon}$, $T_{\pentagon}$ on $M_{\pentagon}$)

$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

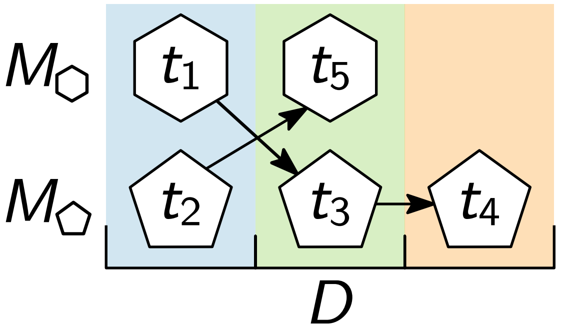
$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$

Reduction (Idea):

- tasks as vertices
- timeslots as colors
- arcs enforce partial order



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices

$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
    - start t^- and end vertex t^+ for each task t

$T_{\text{⬡}} = \{t_1, t_5\}$

$T_{\text{⬠}} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$

paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
  - start t^- and end vertex t^+ for each task t

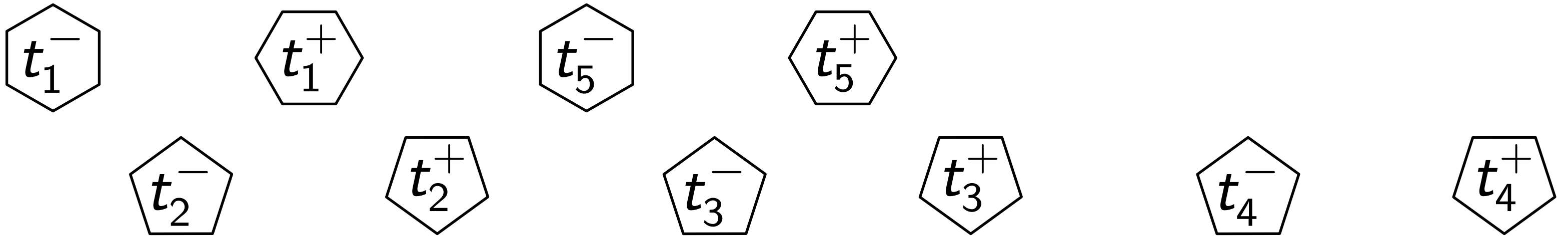
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
    - start t^- and end vertex t^+ for each task t
    - arc (t^-, t^+) for each task t

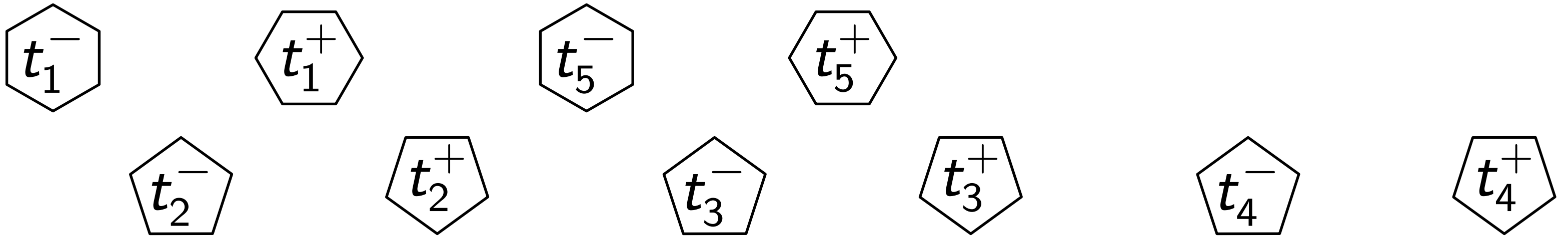
$T_{⬡} = \{t_1, t_5\}$

$T_{⬠} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



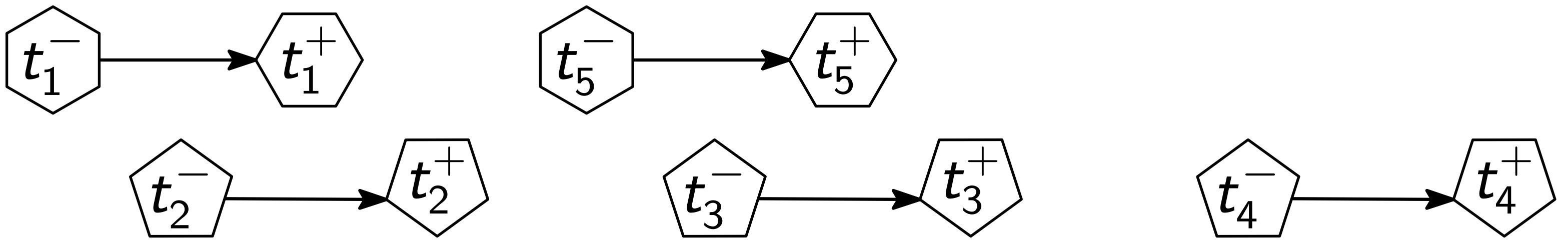
paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
  - start t^- and end vertex t^+ for each task t
  - arc (t^-, t^+) for each task t

$T_{\hexagon} = \{t_1, t_5\}$
$T_{\pentagon} = \{t_2, t_3, t_4\}$
$t_1 \prec t_3 \prec t_4$
$t_2 \prec t_5$
$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
  - start t^- and end vertex t^+ for each task t
  - arc (t^-, t^+) for each task t
  - arc (t_i^+, t_j^-) for $t_i \prec t_j$

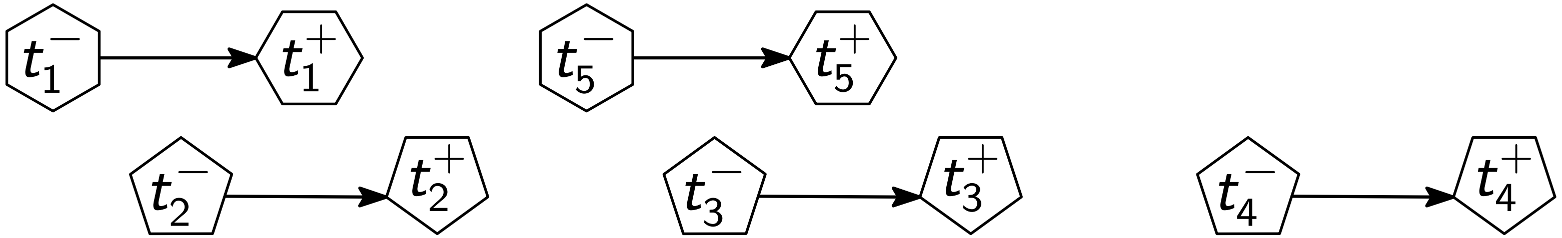
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
  - start t^- and end vertex t^+ for each task t
  - arc (t^-, t^+) for each task t
  - arc (t_i^+, t_j^-) for $t_i \prec t_j$

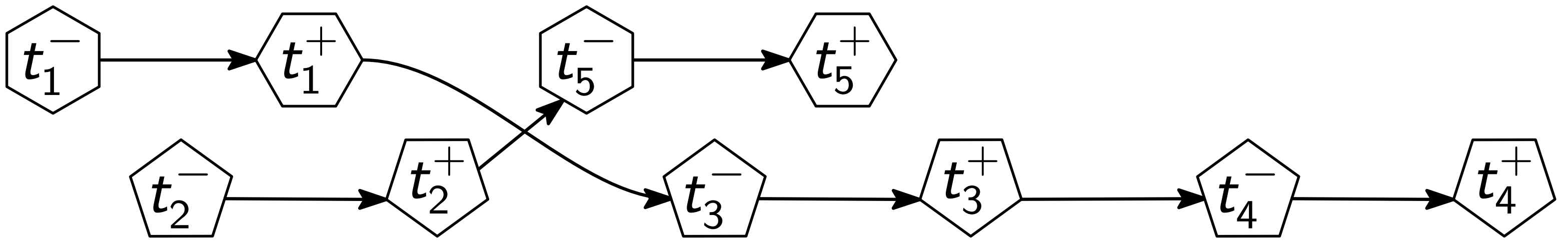
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices

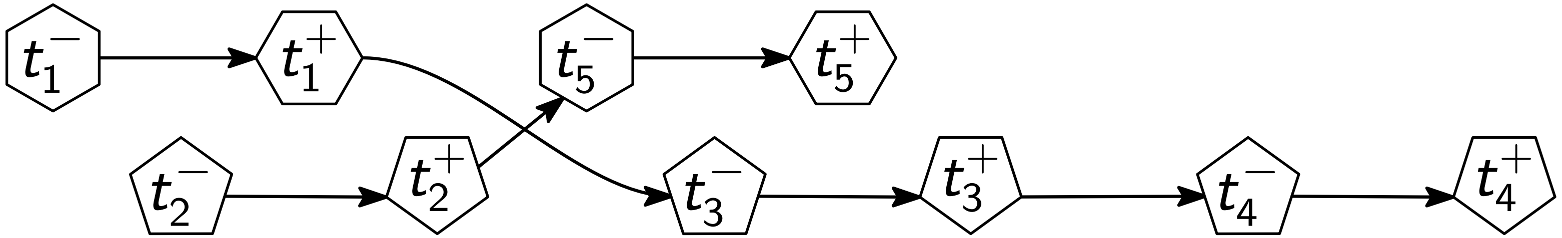
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors

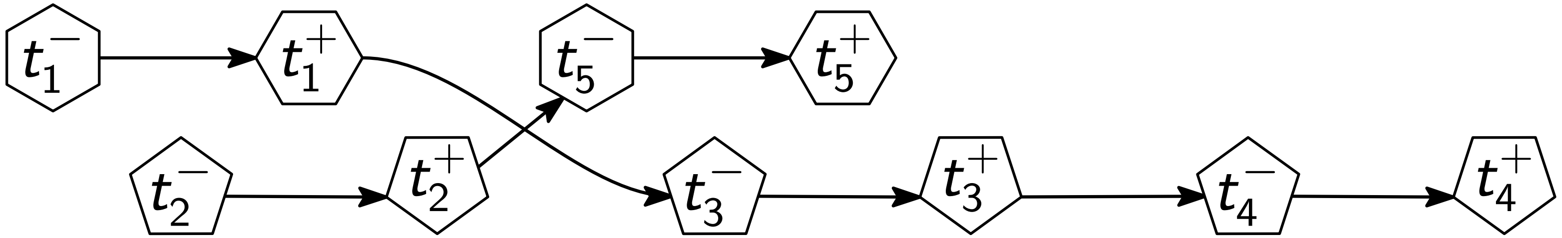
$T_{\text{⬡}} = \{t_1, t_5\}$

$T_{\text{⬠}} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
    - four colors per time unit

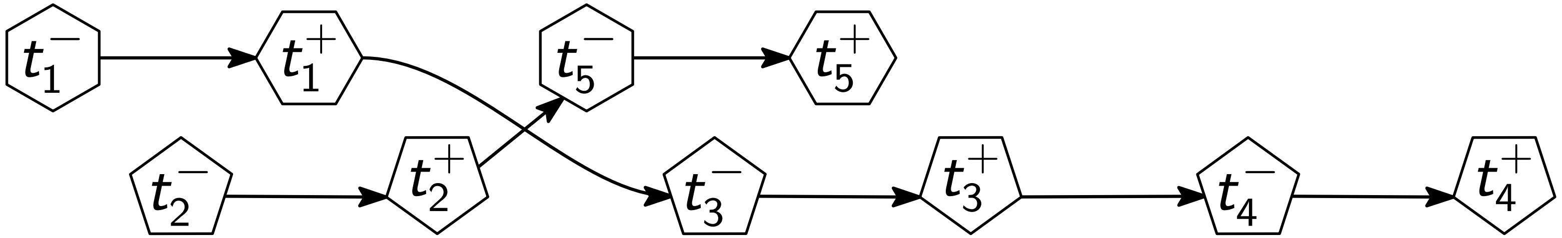
$T_{⬡} = \{t_1, t_5\}$

$T_{⬠} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
    - four colors per time unit

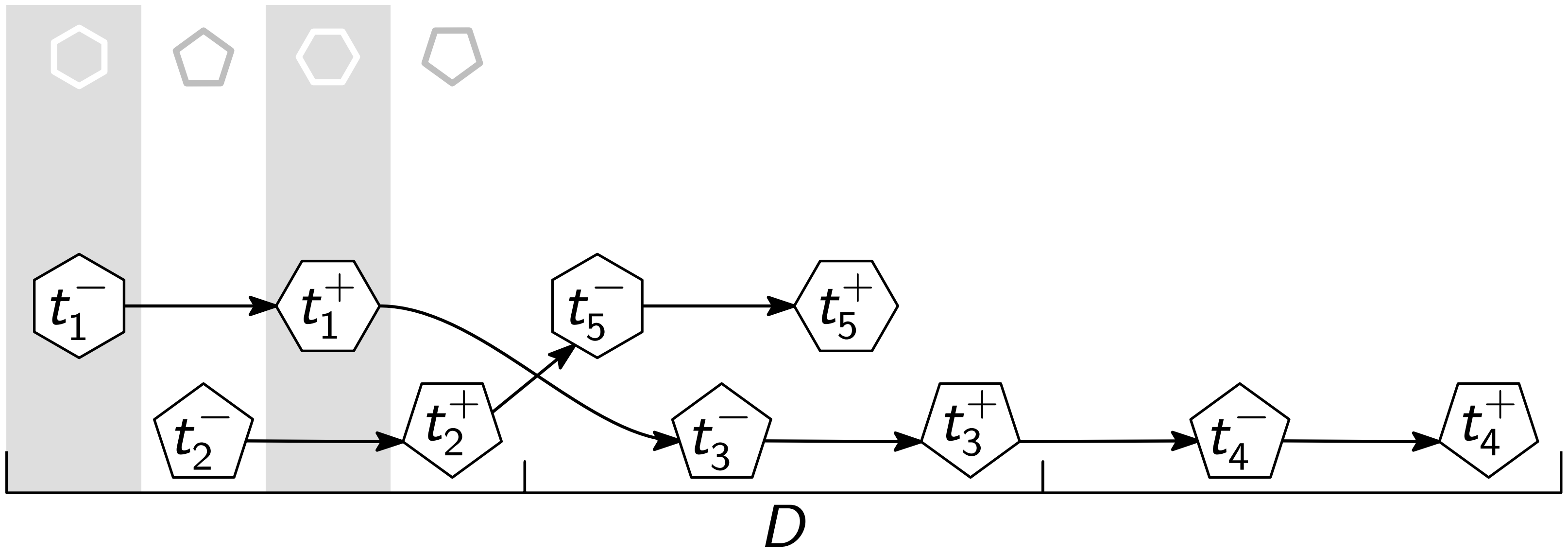
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
  - four colors per time unit

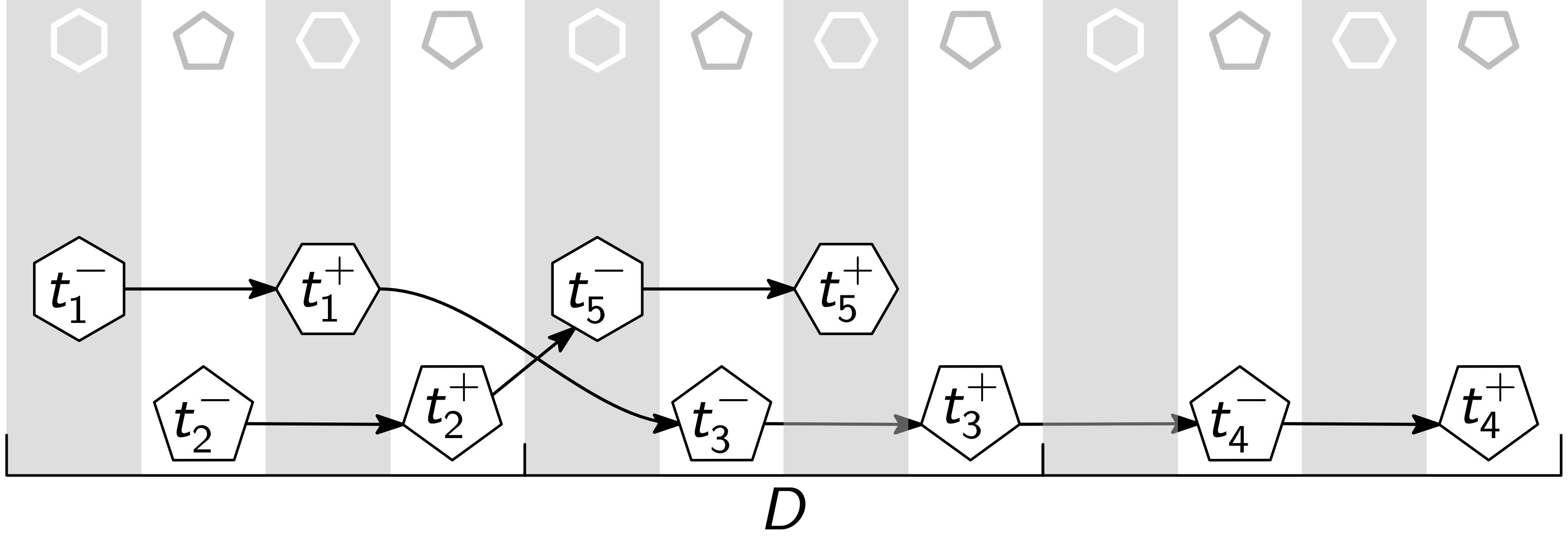
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
  - four colors per time unit

$4D$-colors in total

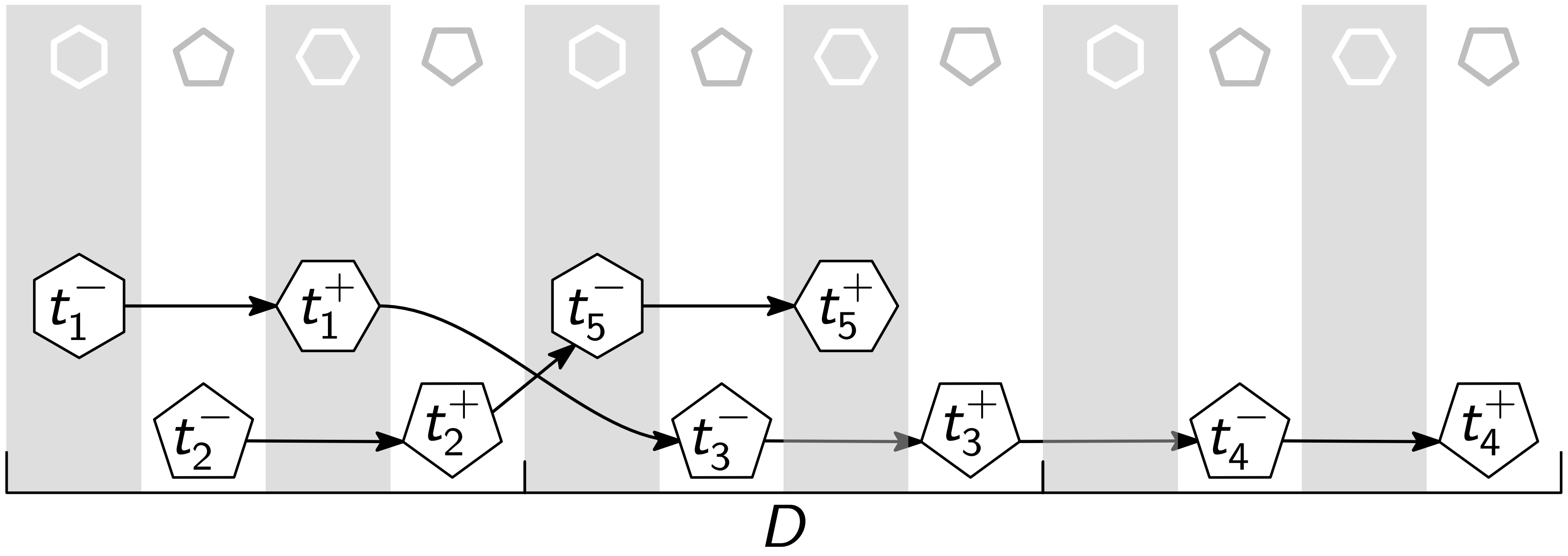
$T_{⬡} = \{t_1, t_5\}$

$T_{⬠} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
  - four colors per time unit
  - colors enforced by directed path of length $4D$

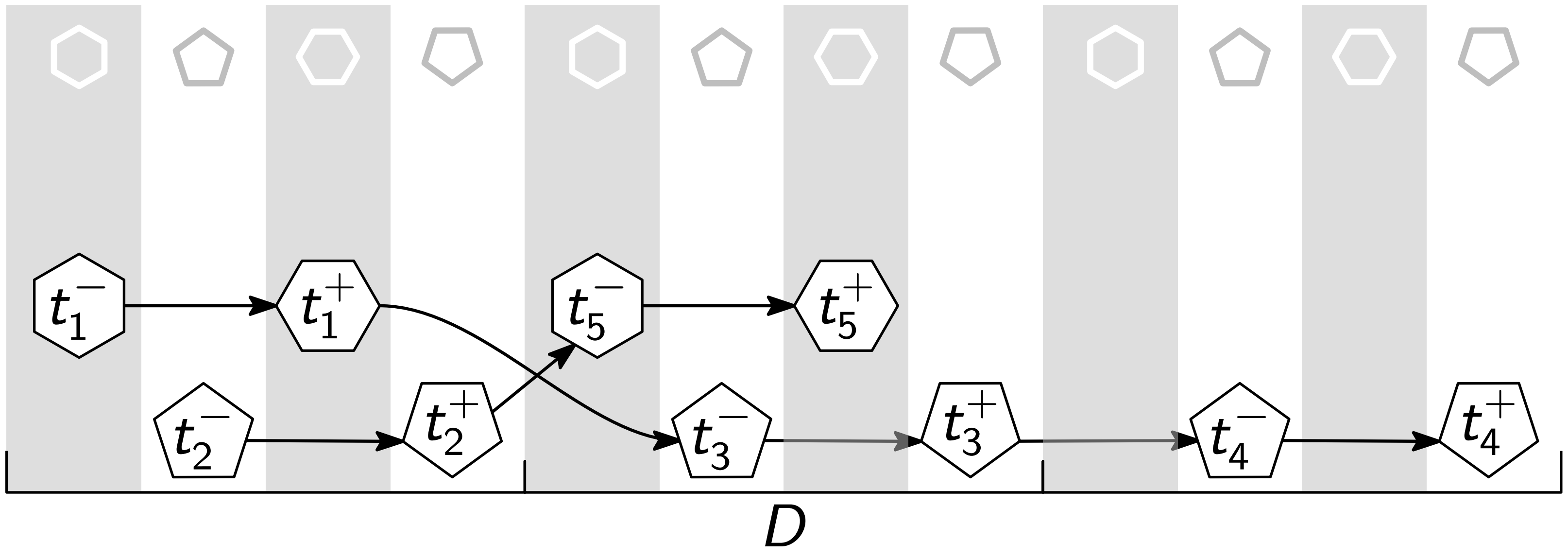
$T_{⬡} = \{t_1, t_5\}$

$T_{⬠} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
  - four colors per time unit
  - colors enforced by directed path of length $4D$

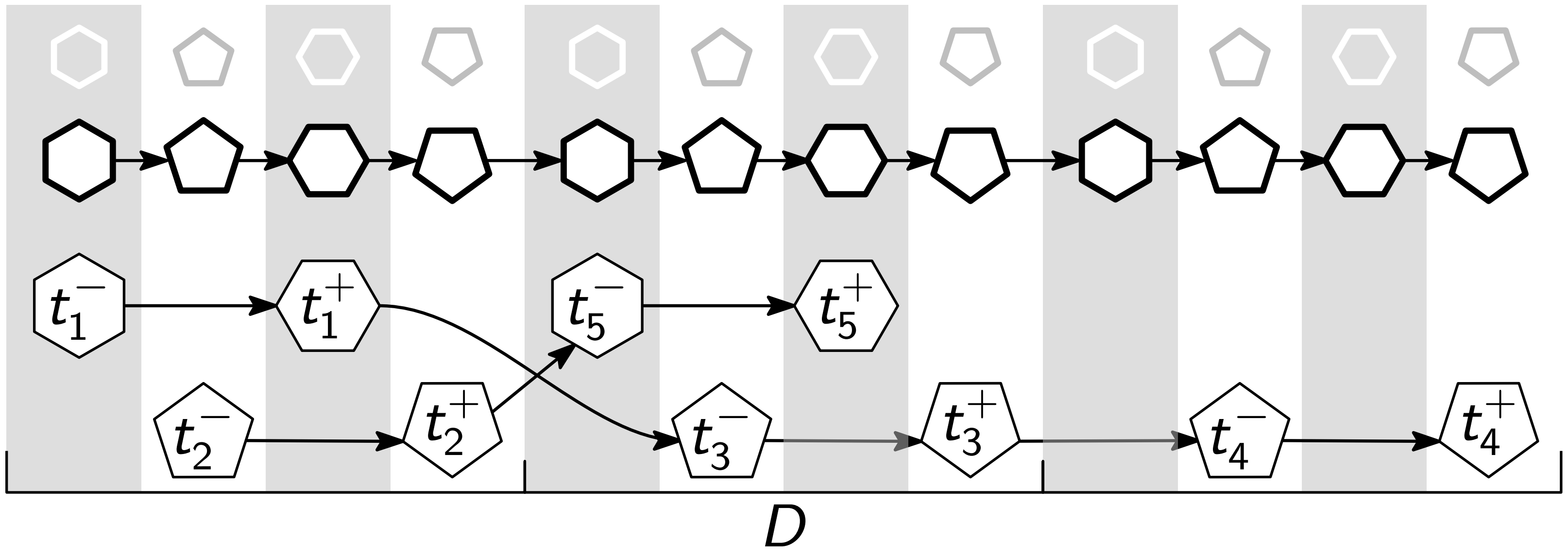
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
  - four colors per time unit
  - colors enforced by directed path of length $4D$

unique $4D$-coloring

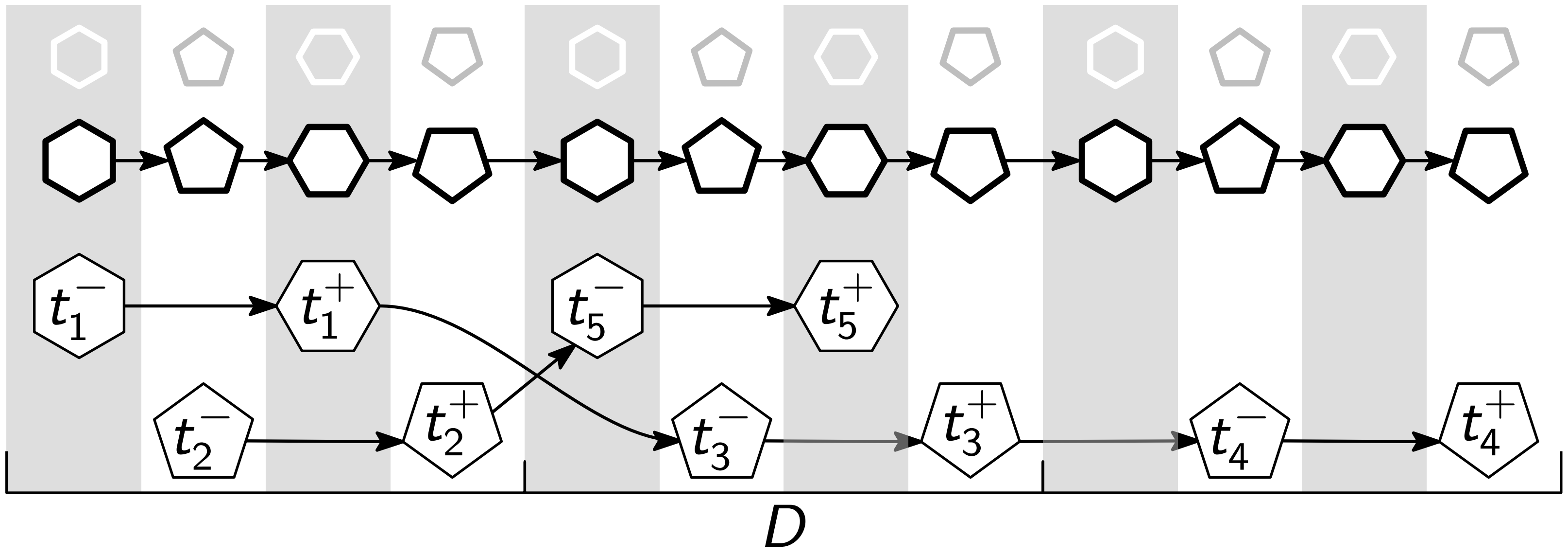
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
    - four colors per time unit
    - colors enforced by directed path of length $4D$

unique $4D$-coloring

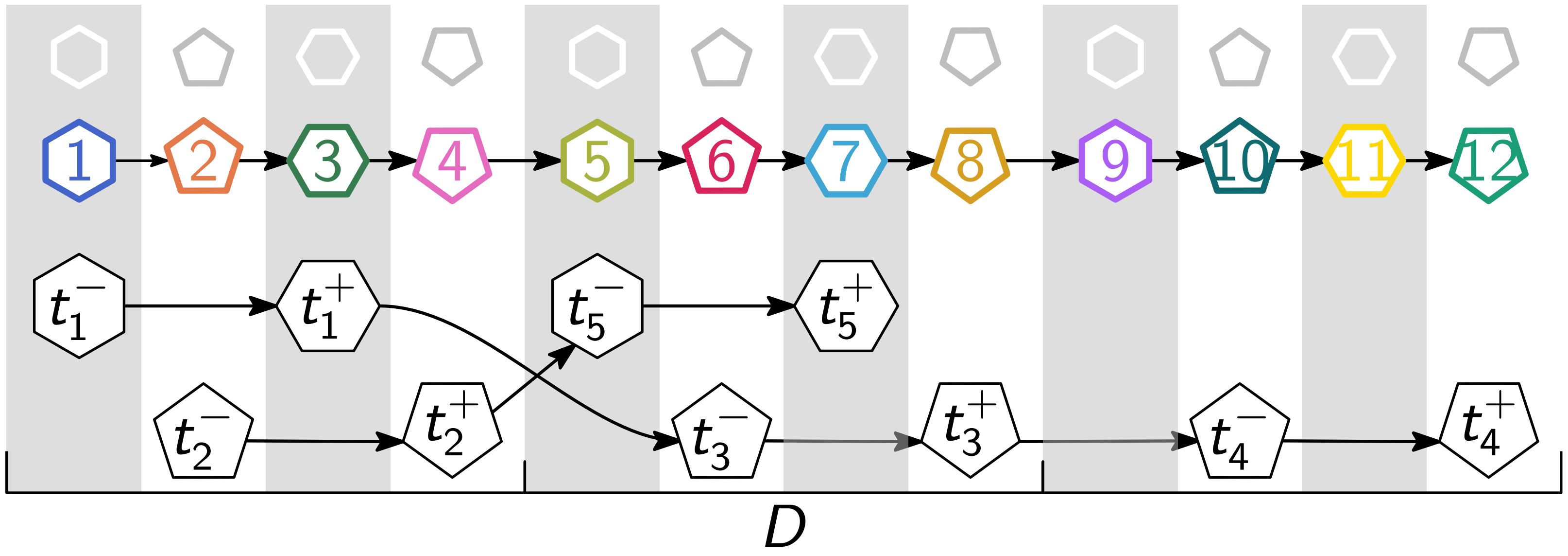
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
    - four colors per time unit
    - colors enforced by directed path of length $4D$

unique $4D$-coloring

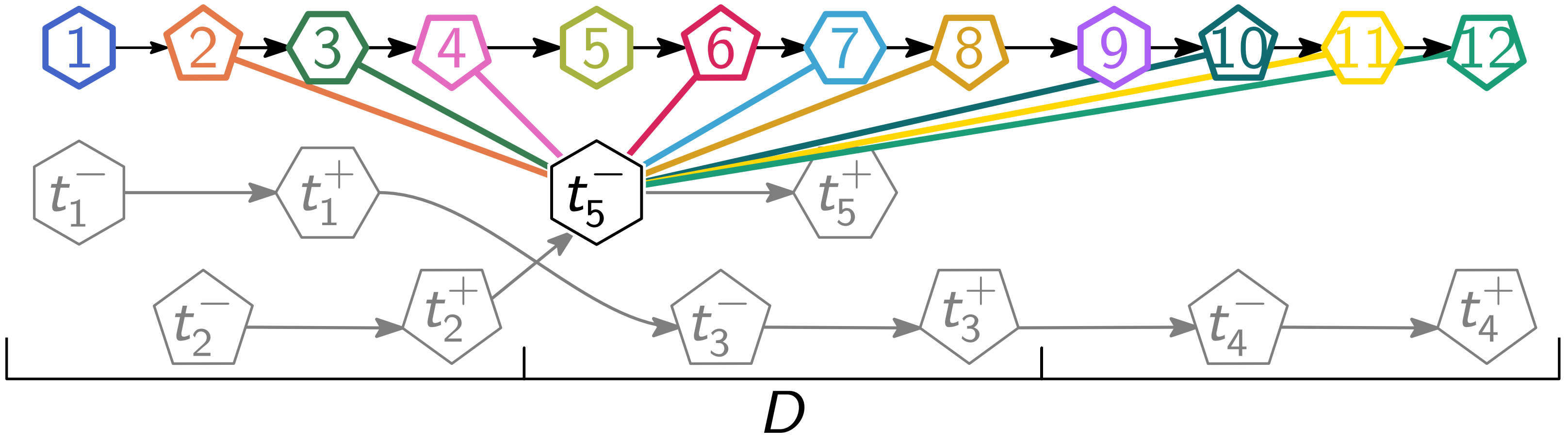
$T_{\text{⬡}} = \{t_1, t_5\}$

$T_{\text{⬠}} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
  - four colors per time unit
  - colors enforced by directed path of length $4D$

unique $4D$-coloring

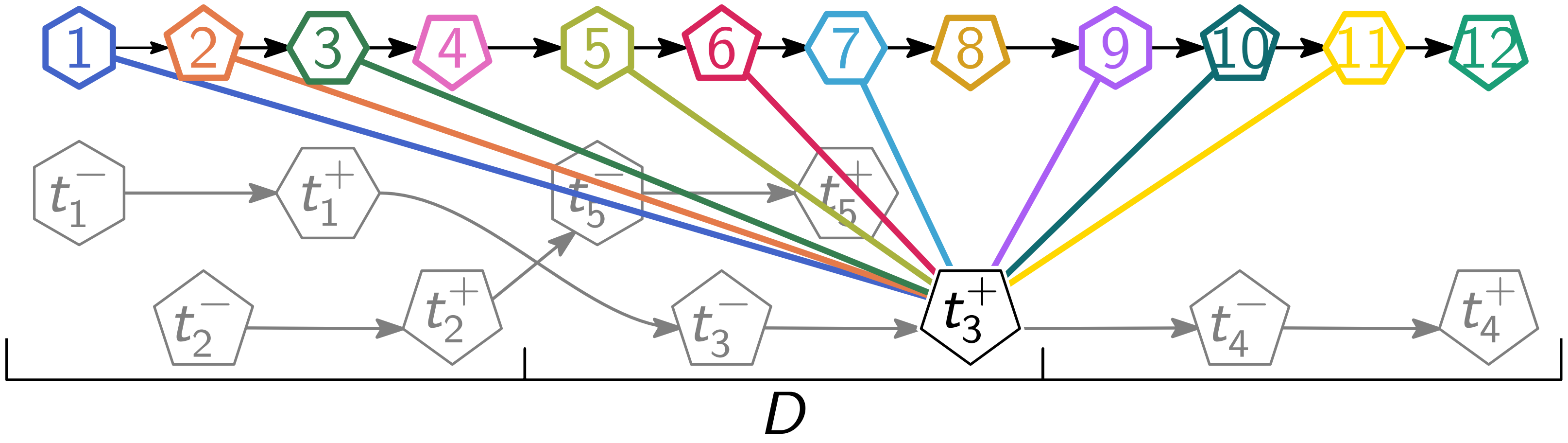
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
    - four colors per time unit
    - colors enforced by directed path of length $4D$

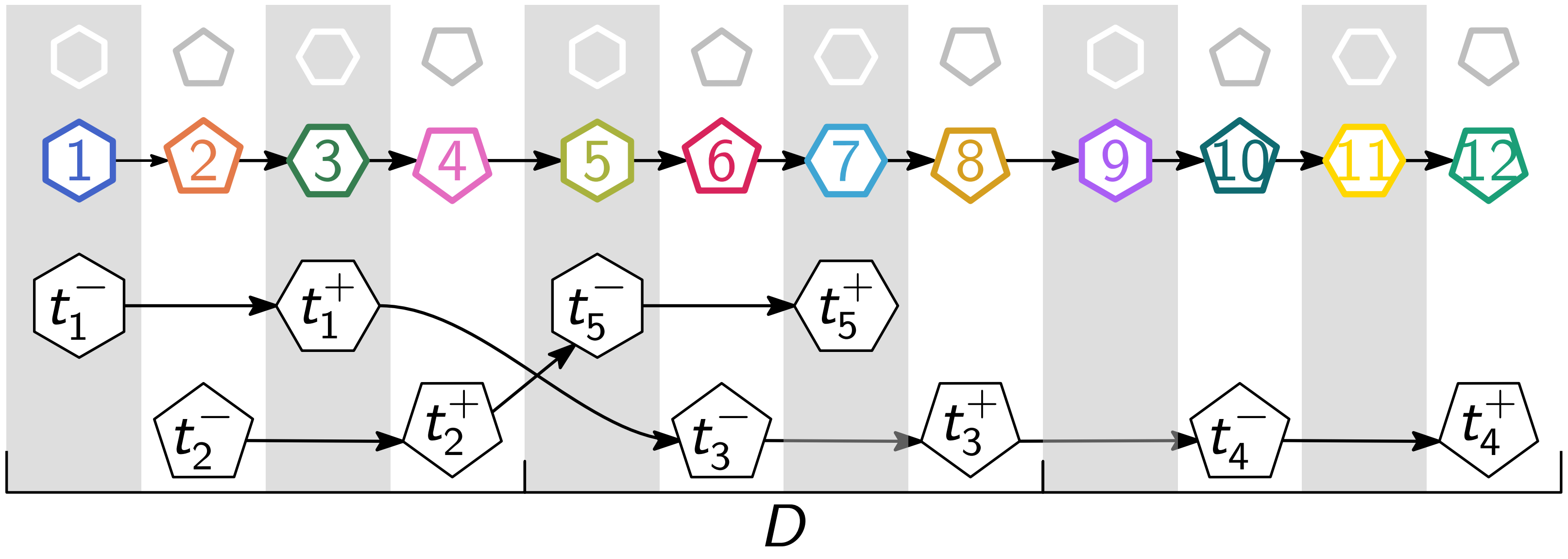
$T_{\text{⬡}} = \{t_1, t_5\}$

$T_{\text{⬠}} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
    - four colors per time unit
    - colors enforced by directed path of length $4D$
    - task vertices pairwise connected

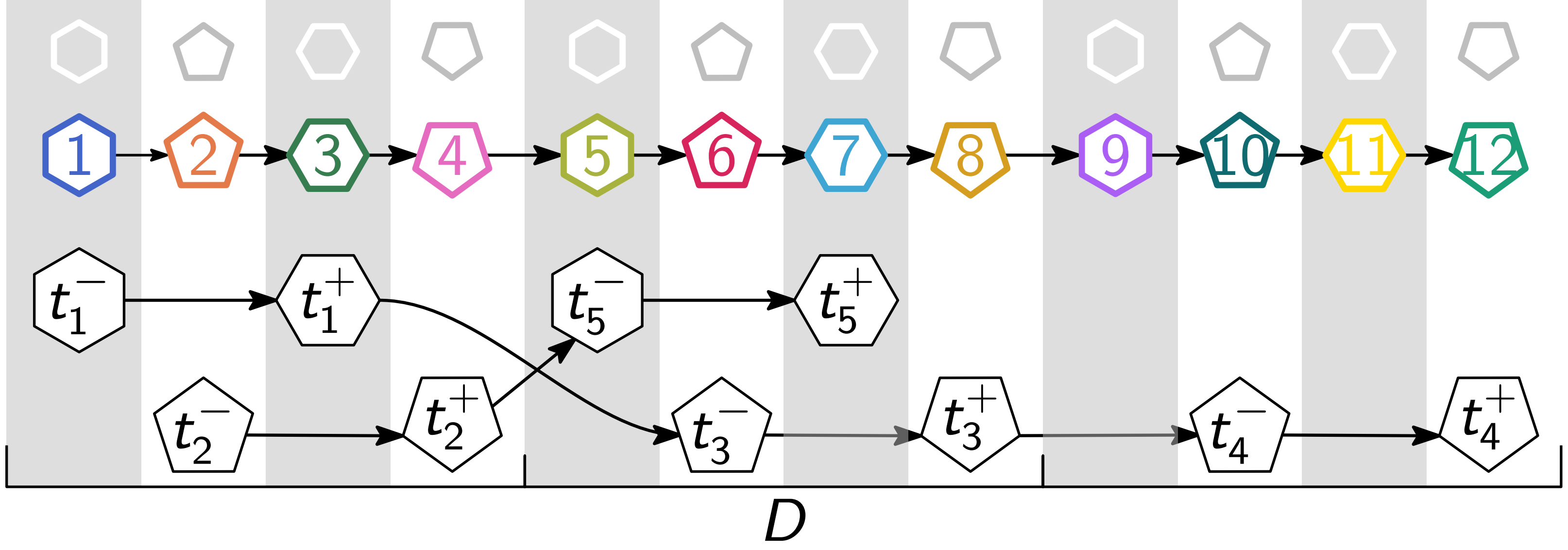
$T_{\text{⬡}} = \{t_1, t_5\}$

$T_{\text{⬠}} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
  - four colors per time unit
  - colors enforced by directed path of length $4D$
  - task vertices pairwise connected
  - vertices of directed path pairwise connected

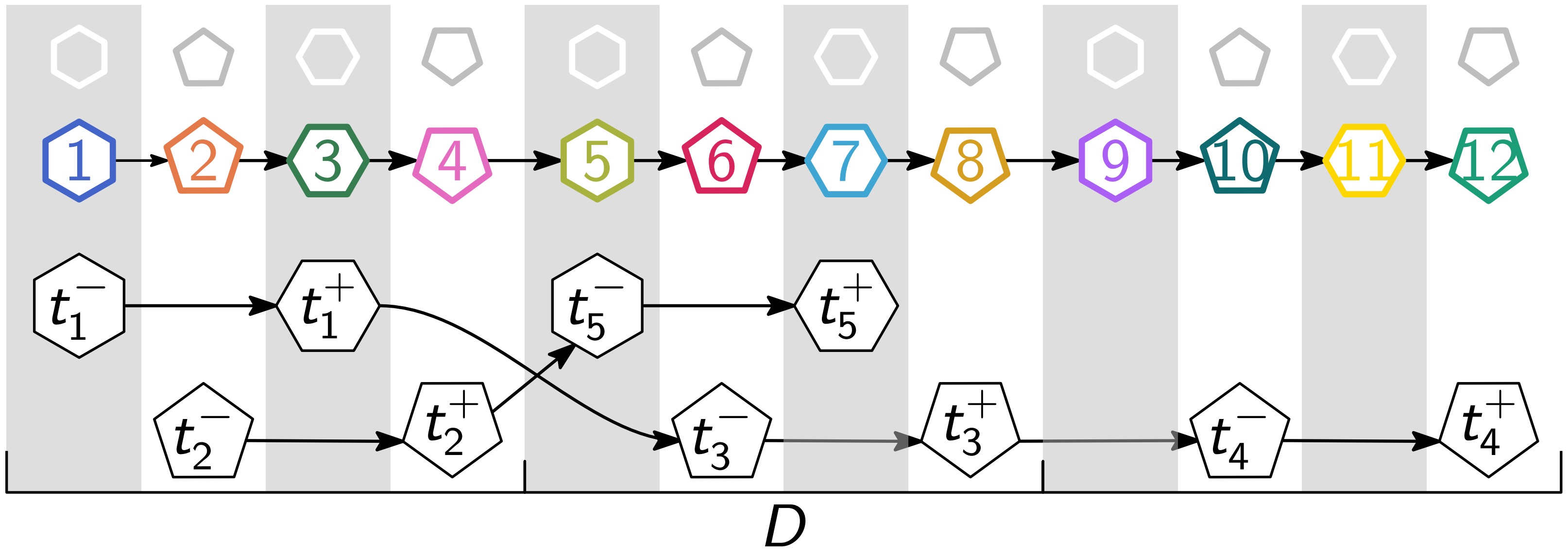
$T_{⬡} = \{t_1, t_5\}$

$T_{⬠} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors

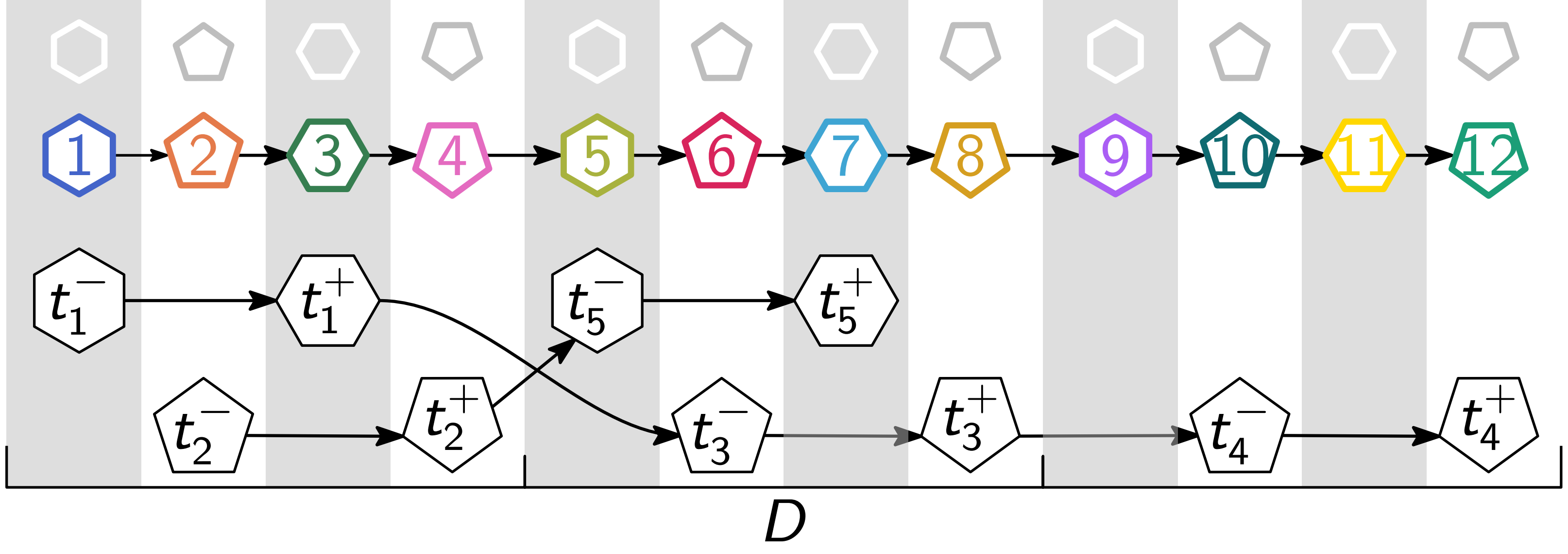
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
- constant neighborhood diversity

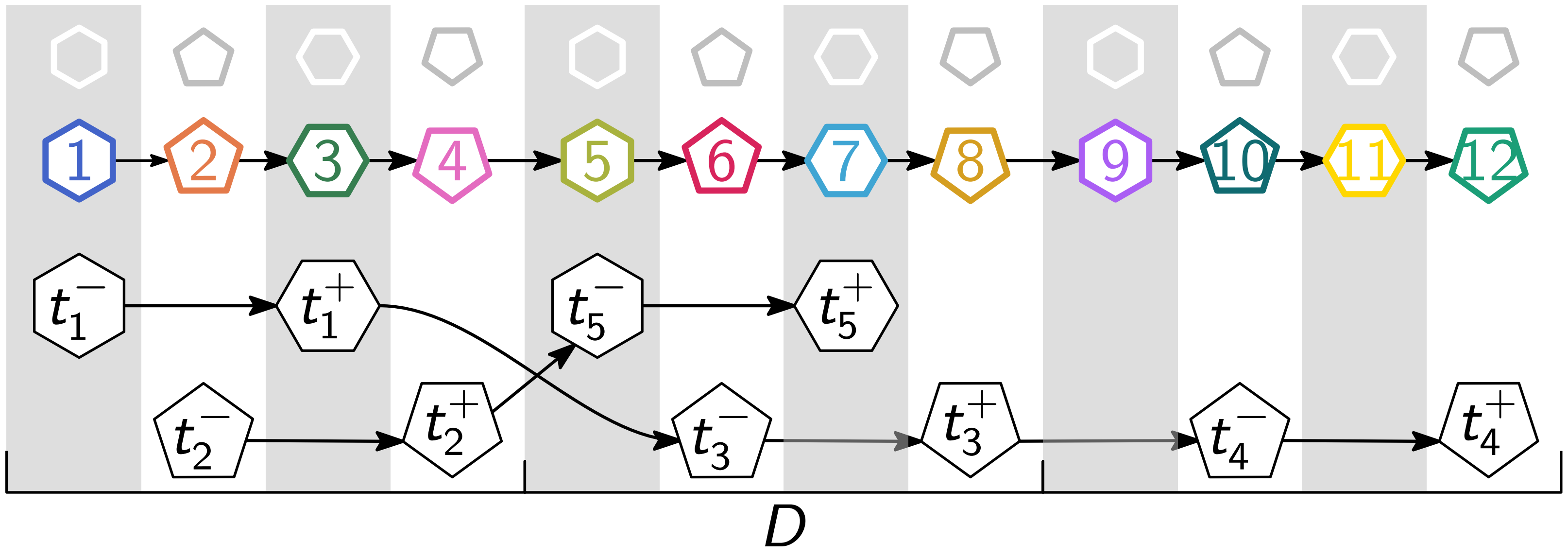
$T_{⬡} = \{t_1, t_5\}$

$T_{⬠} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
- constant neighborhood diversity
  - 4 types of task vertices

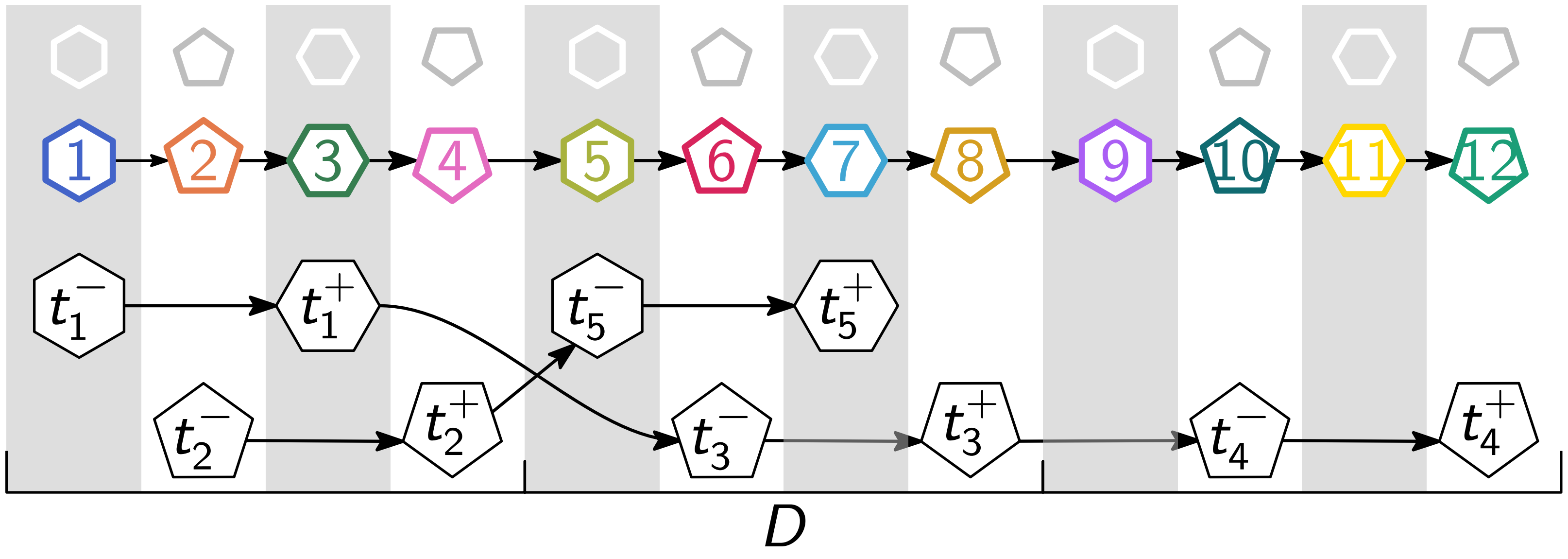
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
- constant neighborhood diversity
    - 4 types of task vertices
    - 4 types of path vertices

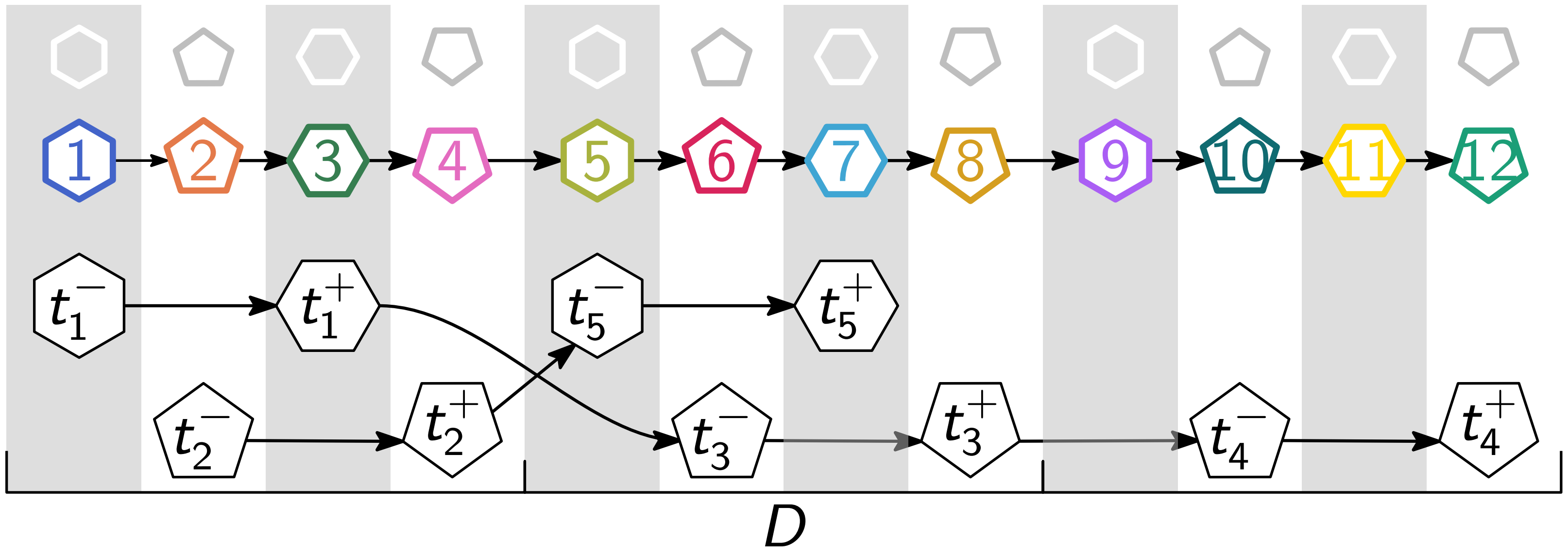
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
- constant neighborhood diversity
- graph has proper $4D$-coloring ⇔ exists feasible schedule of length D

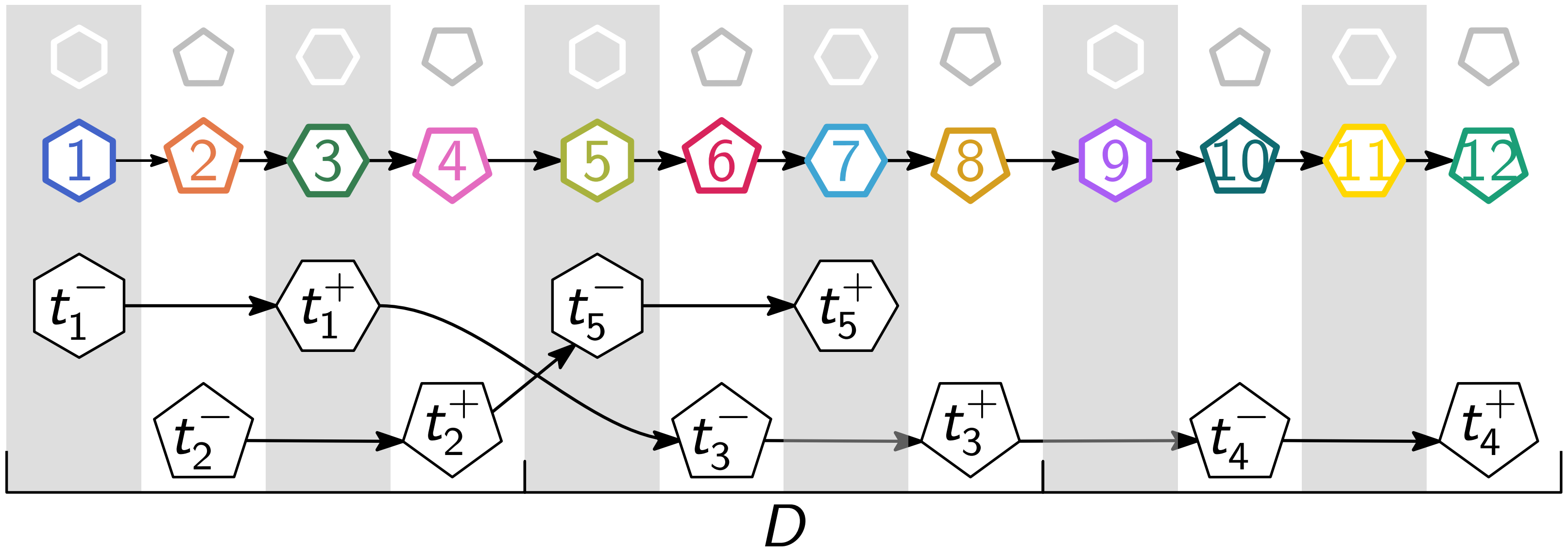
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
- constant neighborhood diversity
- graph has proper $4D$-coloring ⇔ exists feasible schedule of length D

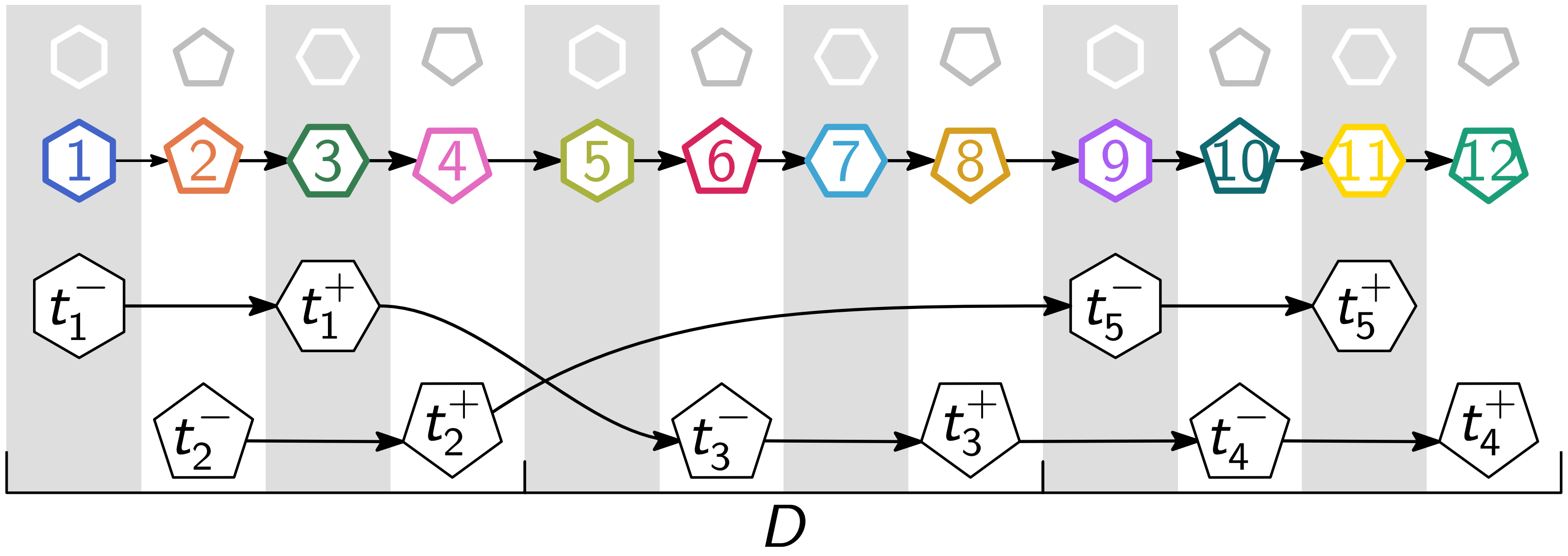
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



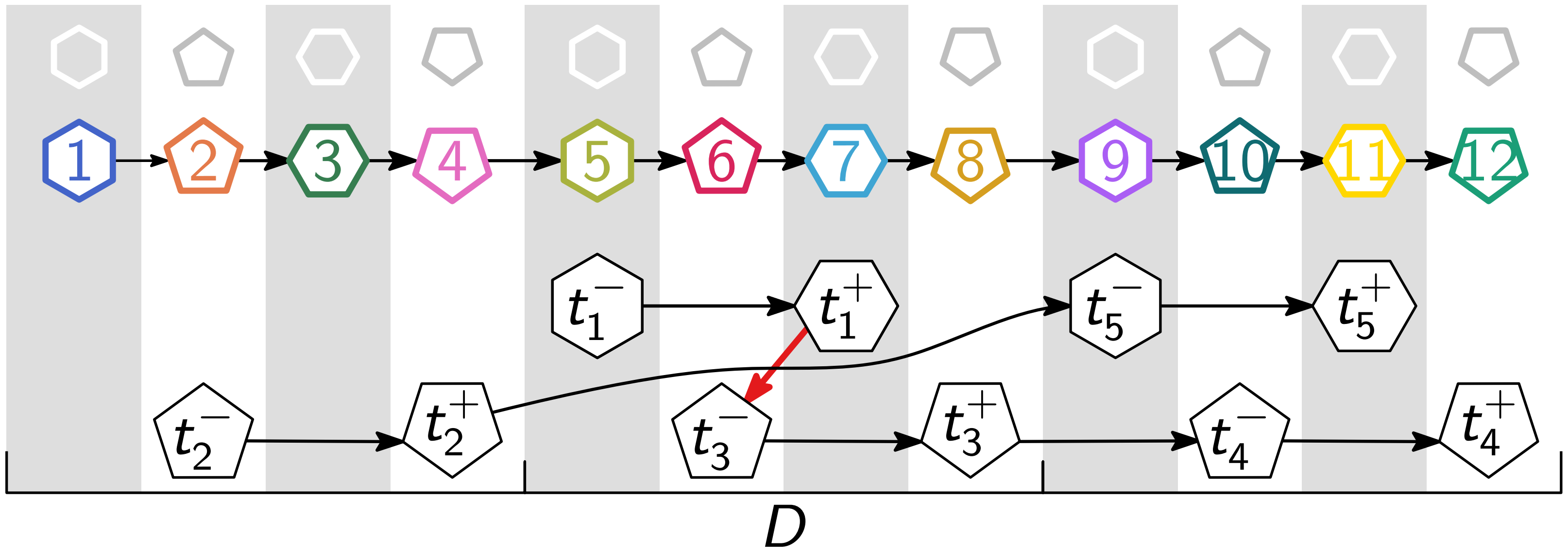
paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
- constant neighborhood diversity
- graph has proper $4D$-coloring ⇔ exists feasible schedule of length D

$$T_{\hexagon} = \{t_1, t_5\}$$
$$T_{\pentagon} = \{t_2, t_3, t_4\}$$
$$t_1 \prec t_3 \prec t_4$$
$$t_2 \prec t_5$$
$$D = 3$$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
- constant neighborhood diversity
- graph has proper $4D$-coloring ⇔ exists feasible schedule of length D

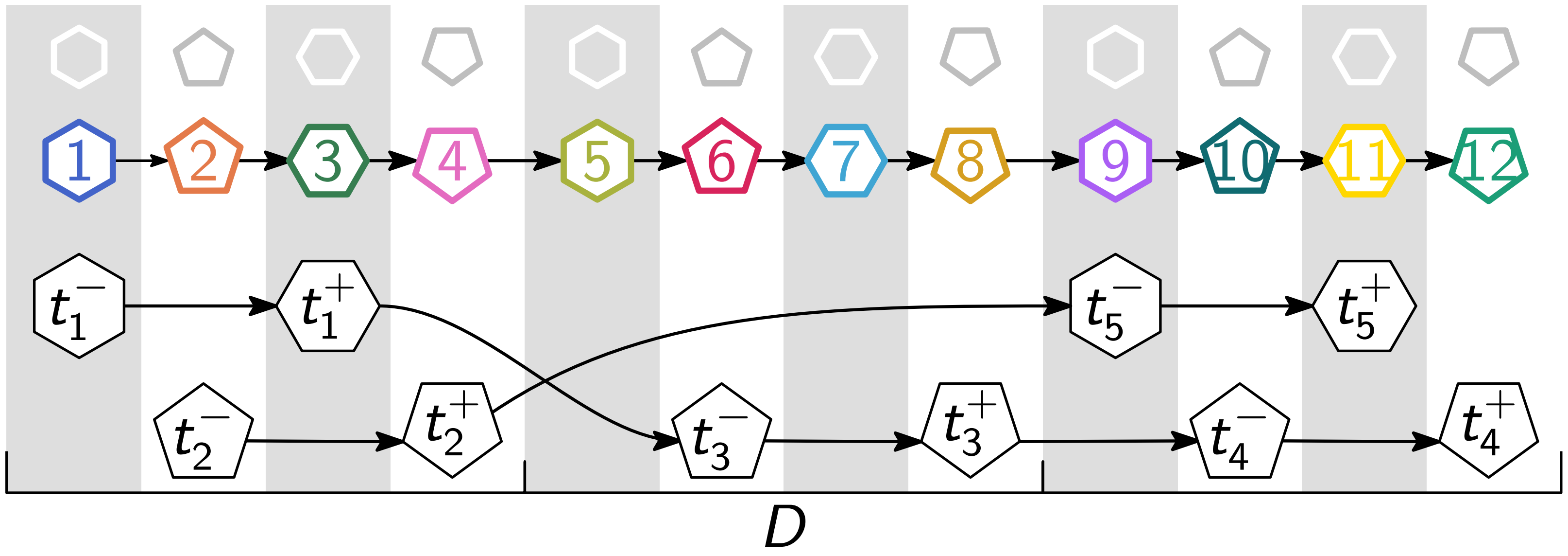
$T_{\text{⬡}} = \{t_1, t_5\}$

$T_{\text{⬠}} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
- constant neighborhood diversity
- graph has proper $4D$-coloring $\Leftrightarrow$ exists feasible schedule of length D

$\implies$ MixedColoring NP-hard for constant neighborhood diversity

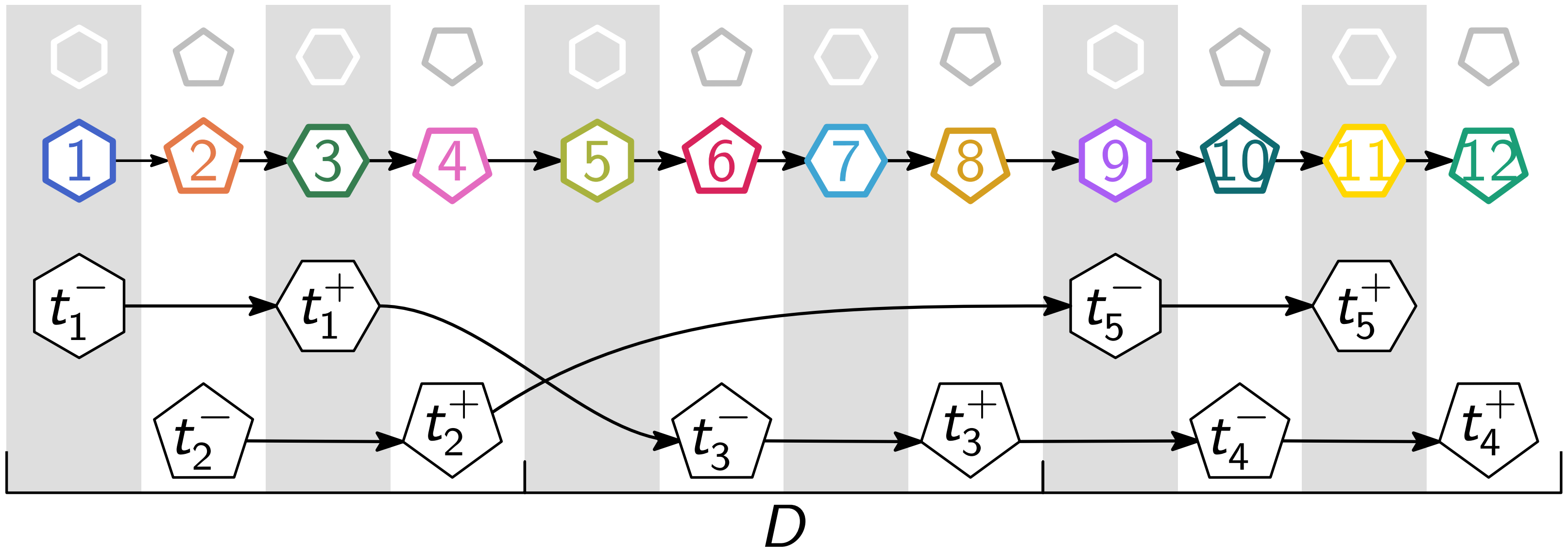
$T_{\text{⬡}} = \{t_1, t_5\}$

$T_{\text{⬠}} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



paraNP-Hardness Neighborhood Diversity

Goal: show NP-hardness for constant neighborhood diversity

Reduction:

- tasks as vertices
- timeslots as colors
- constant neighborhood diversity
- graph has proper $4D$-coloring $\Leftrightarrow$ exists feasible schedule of length D

$\Longrightarrow$ MixedColoring paraNP-hard w.r.t. neighborhood diversity

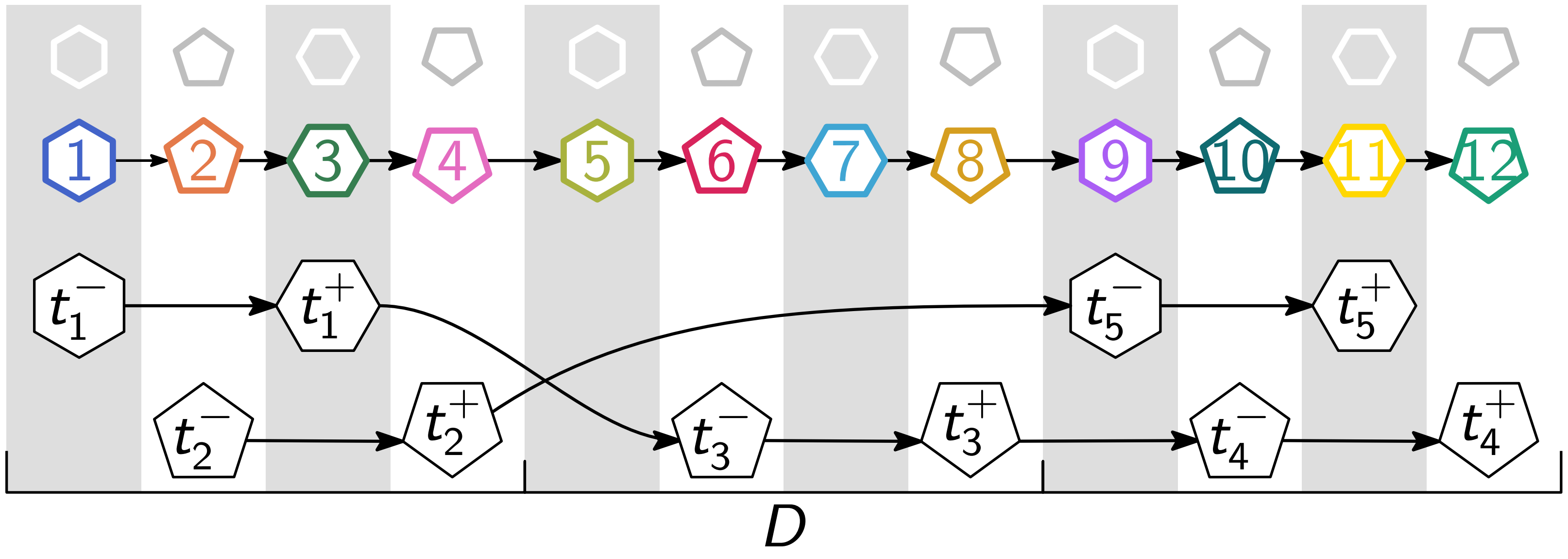
$T_{\hexagon} = \{t_1, t_5\}$

$T_{\pentagon} = \{t_2, t_3, t_4\}$

$t_1 \prec t_3 \prec t_4$

$t_2 \prec t_5$

$D = 3$



FPT Mixed Neighborhood Diversity

FPT Mixed Neighborhood Diversity

Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

FPT Mixed Neighborhood Diversity

Theorem

MIXEDCOLORING is FPT w.r.t. mixed neighborhood diversity

Recall:

- vertices of each type induce clique or independent set
- vertices in type inducing independent set can receive same color

FPT Mixed Neighborhood Diversity

Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

Recall:

- vertices of each type induce clique or independent set
- vertices in type inducing independent set can receive same color

Assumption: each type induces clique

FPT Mixed Neighborhood Diversity

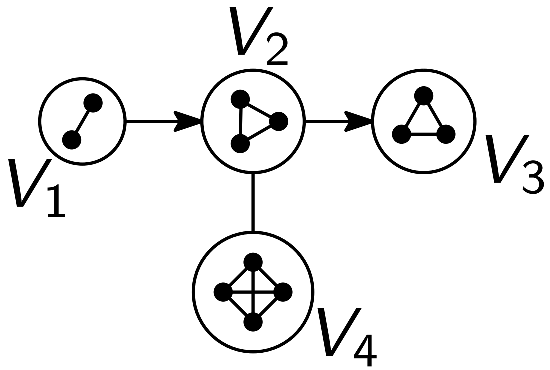
Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

Recall:

- vertices of each type induce clique or independent set
- vertices in type inducing independent set can receive same color

Assumption: each type induces clique



FPT Mixed Neighborhood Diversity

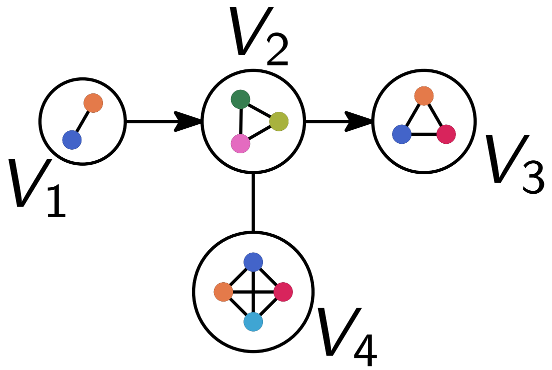
Theorem

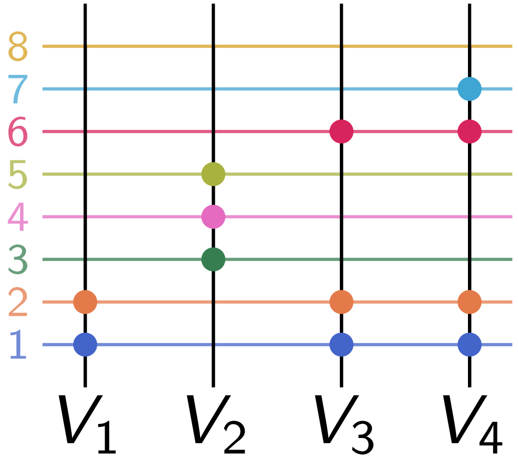
MixedColoring is FPT w.r.t. mixed neighborhood diversity

Recall:

- vertices of each type induce clique or independent set
- vertices in type inducing independent set can receive same color

Assumption: each type induces clique





FPT Mixed Neighborhood Diversity

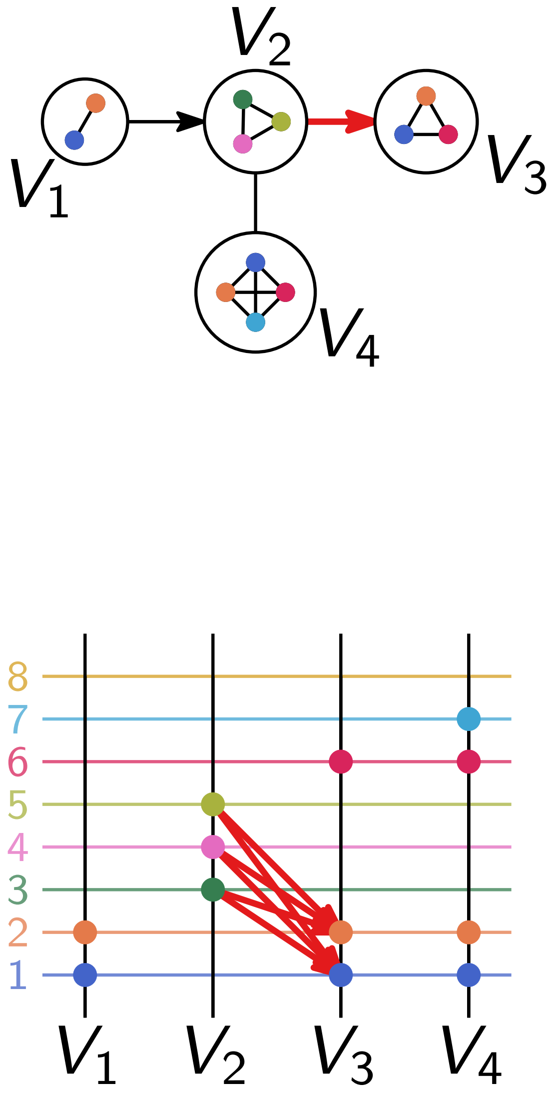
Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

Recall:

- vertices of each type induce clique or independent set
- vertices in type inducing independent set can receive same color

Assumption: each type induces clique



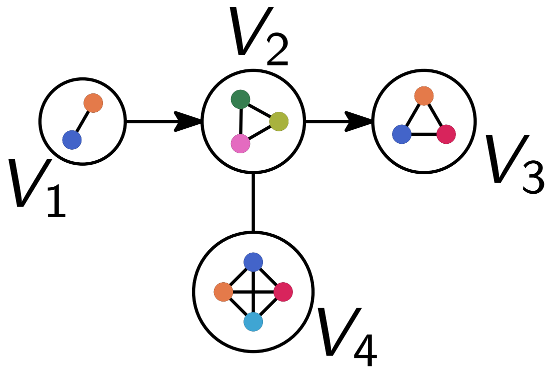
FPT Mixed Neighborhood Diversity

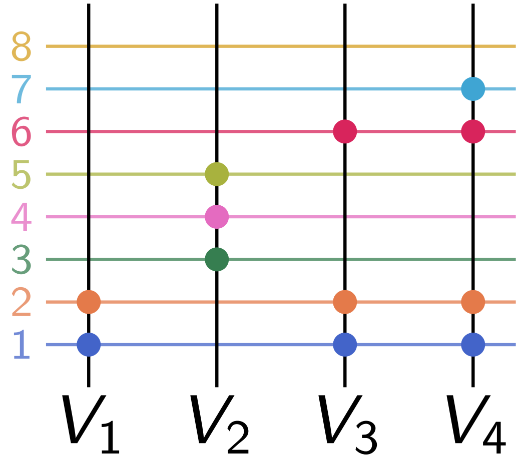
Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

Lemma

coloring c respects arcs if $\forall (V_i, V_j) \in A\colon c^{+}(V_i) \leq c^{-}(V_j)$





FPT Mixed Neighborhood Diversity

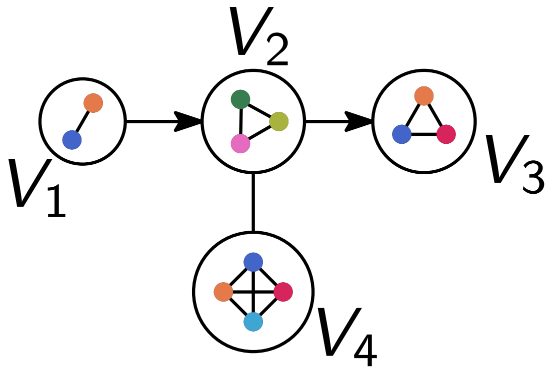
Theorem
MixedColoring is FPT w.r.t. mixed neighborhood diversity

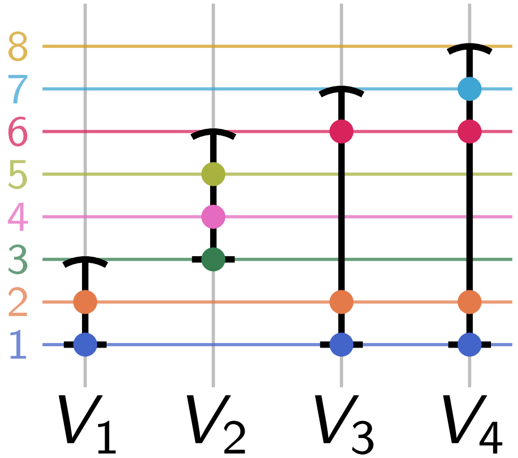
Lemma

coloring c respects arcs if $\forall (V_i, V_j) \in A\colon c^{+}(V_i) \leq c^{-}(V_j)$

largest color in $V_i + 1$

smallest color in V_j





FPT Mixed Neighborhood Diversity

Theorem

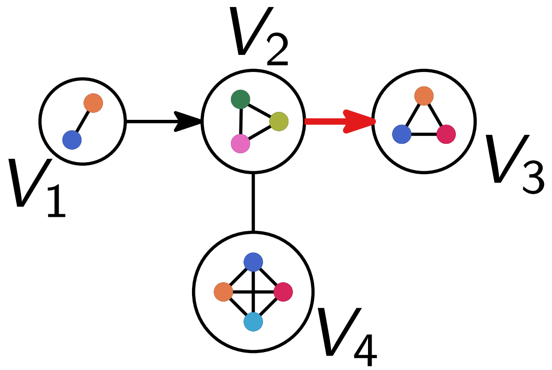
MixedColoring is FPT w.r.t. mixed neighborhood diversity

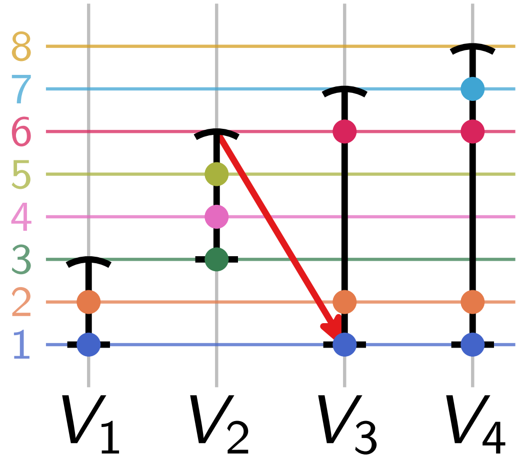
Lemma

coloring c respects arcs if $\forall(V_i, V_j) \in A$: $c^+(V_i) \leq c^-(V_j)$

largest color in $V_i + 1$

smallest color in V_j





FPT Mixed Neighborhood Diversity

Theorem

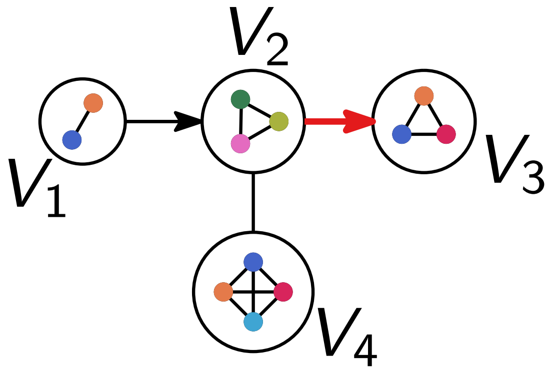
MIXEDCOLORING is FPT w.r.t. mixed neighborhood diversity

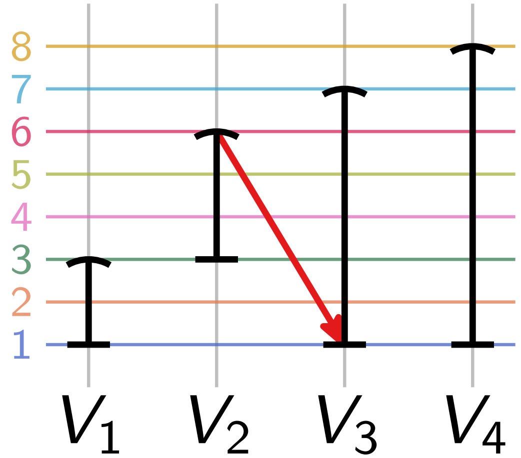
Lemma

coloring c respects arcs if $\forall (V_i, V_j) \in A$: $c^{+}(V_i) \leq c^{-}(V_j)$

largest color in $V_i + 1$

smallest color in V_j





FPT Mixed Neighborhood Diversity

Theorem

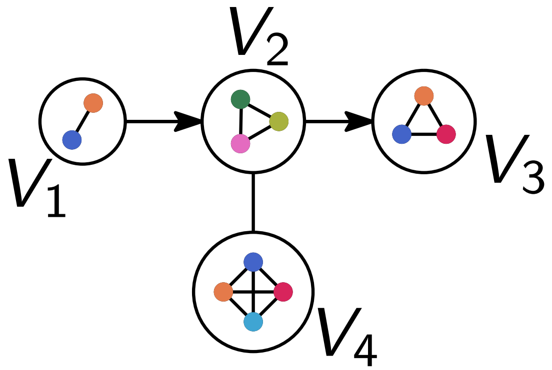
MixedColoring is FPT w.r.t. mixed neighborhood diversity

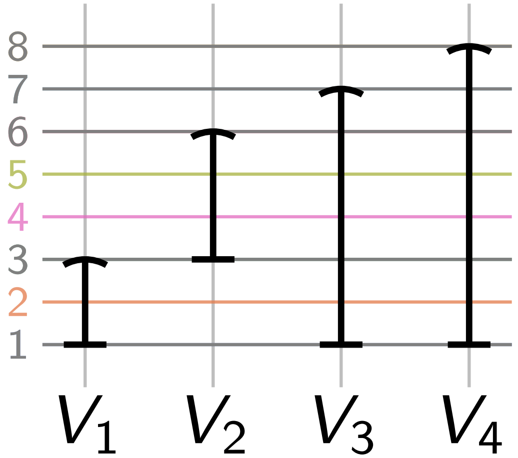
Lemma

coloring c respects arcs if $\forall(V_i, V_j) \in A$: $c^+(V_i) \leq c^-(V_j)$

largest color in $V_i + 1$

smallest color in V_j





FPT Mixed Neighborhood Diversity

Theorem

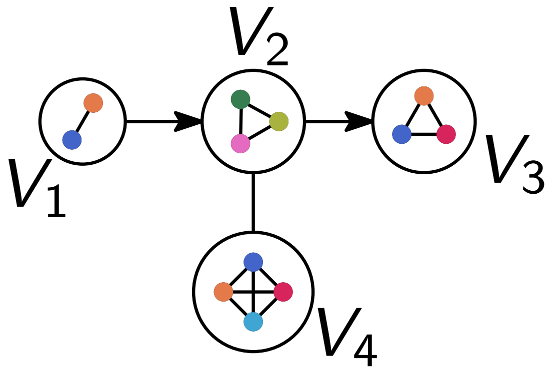
MixedColoring is FPT w.r.t. mixed neighborhood diversity

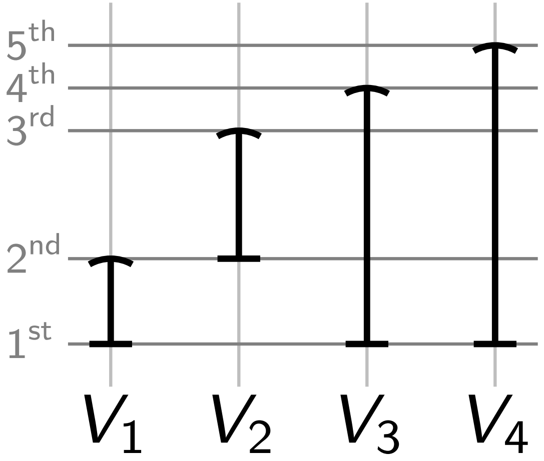
Lemma

coloring c respects arcs if $\forall(V_i, V_j) \in A$: $c^+(V_i) \leq c^-(V_j)$

largest color in $V_i + 1$

smallest color in V_j





FPT Mixed Neighborhood Diversity

Theorem

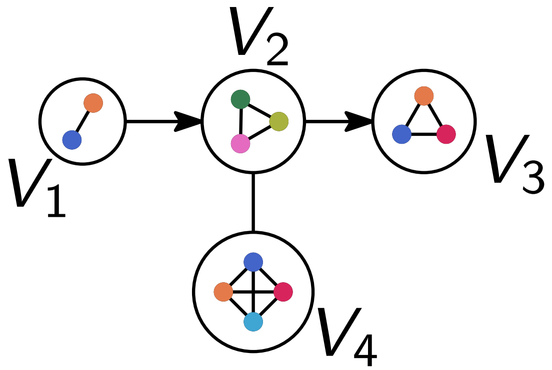
MixedColoring is FPT w.r.t. mixed neighborhood diversity

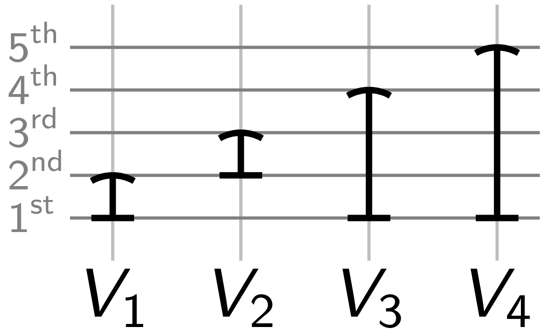
Lemma

coloring c respects arcs if $\forall (V_i, V_j) \in A$: $c^{+}(V_i) \leq c^{-}(V_j)$

largest color in $V_i + 1$

smallest color in V_j





FPT Mixed Neighborhood Diversity

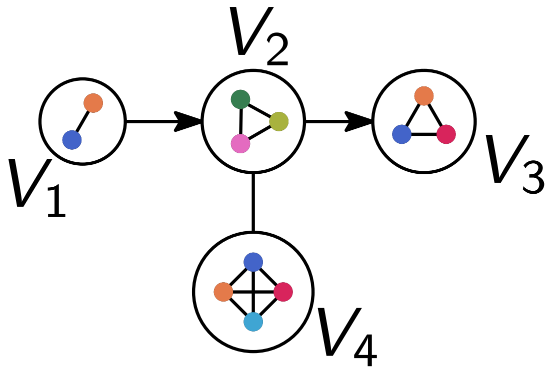
Theorem

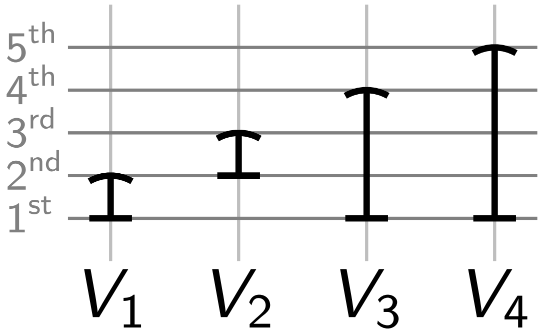
MixedColoring is FPT w.r.t. mixed neighborhood diversity

Lemma

coloring c respects arcs if $\forall (V_i, V_j) \in A$: $c^+(V_i) \leq c^-(V_j)$

Type Preorder





FPT Mixed Neighborhood Diversity

Theorem

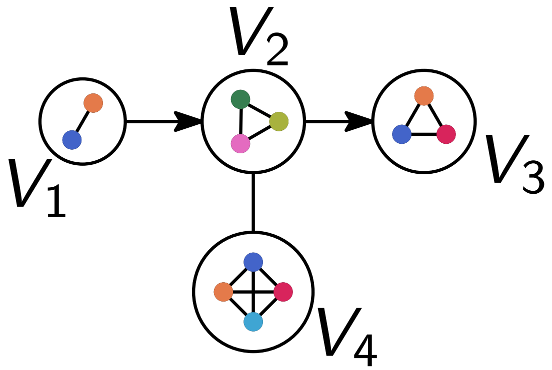
MixedColoring is FPT w.r.t. mixed neighborhood diversity

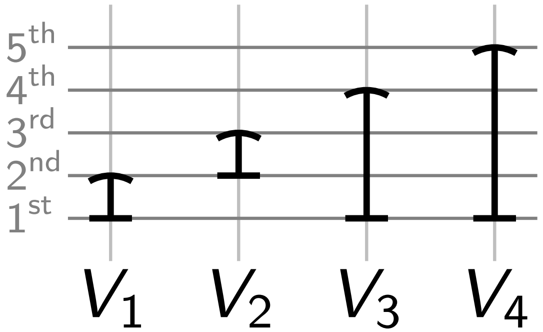
Lemma

coloring c respects arcs if $\forall (V_i, V_j) \in A$: $c^+(V_i) \leq c^-(V_j)$

Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t





FPT Mixed Neighborhood Diversity

Theorem

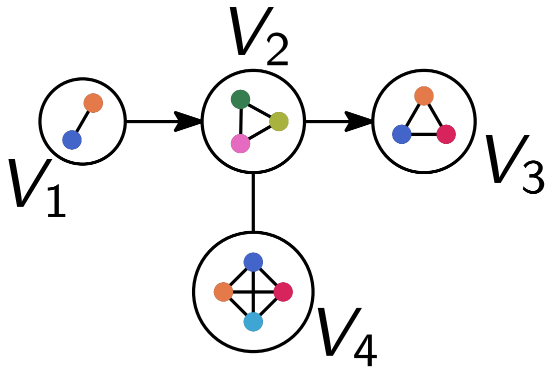
MixedColoring is FPT w.r.t. mixed neighborhood diversity

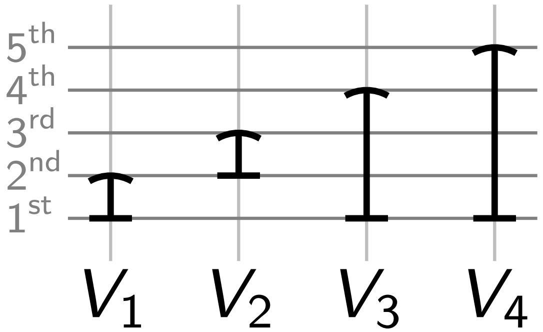
Lemma

coloring c respects arcs if $\forall (V_i, V_j) \in A$: $c^+(V_i) \leq c^-(V_j)$

Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t
- proper if $\forall (V_i, V_j) \in A$: $p^+(V_i) \leq p^-(V_j)$





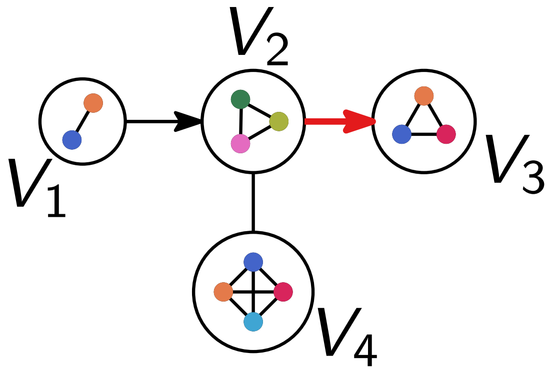
FPT Mixed Neighborhood Diversity

Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

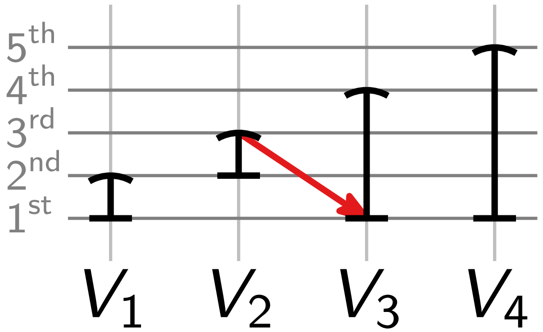
Lemma

coloring c respects arcs if $\forall (V_i, V_j) \in A$: $c^+(V_i) \leq c^-(V_j)$



Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t
- proper if $\forall (V_i, V_j) \in A$: $p^+(V_i) \leq p^-(V_j)$



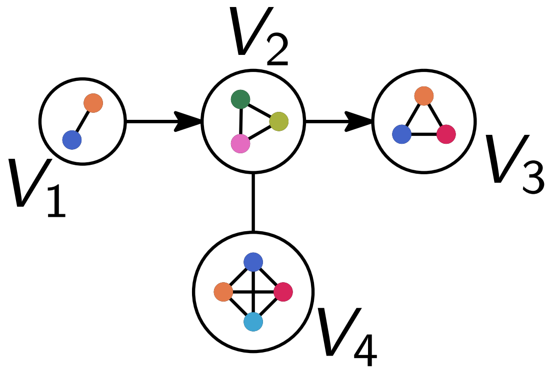
FPT Mixed Neighborhood Diversity

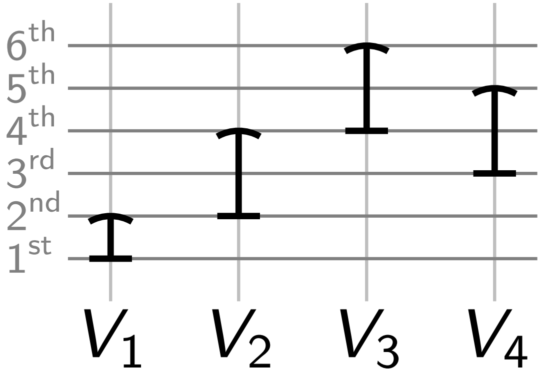
Theorem
MixedColoring is FPT w.r.t. mixed neighborhood diversity

Lemma
coloring c respects arcs if $\forall (V_i, V_j) \in A$: $c^+(V_i) \leq c^-(V_j)$

Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t
- proper if $\forall (V_i, V_j) \in A$: $p^+(V_i) \leq p^-(V_j)$





FPT Mixed Neighborhood Diversity

Theorem

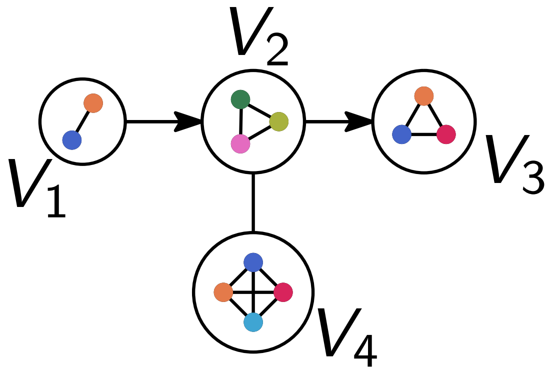
MixedColoring is FPT w.r.t. mixed neighborhood diversity

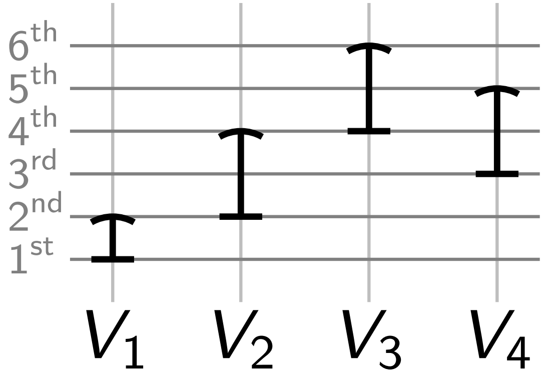
Lemma

coloring c respects arcs if $\forall (V_i, V_j) \in A \colon c^+(V_i) \leq c^-(V_j)$

Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t
- proper if $\forall (V_i, V_j) \in A \colon p^+(V_i) \leq p^-(V_j)$
- coloring c corresponds to preorder p if $\exists c_1 < \cdots < c_\ell \colon c(V_i) \subset [c_{p^-(V_i)}, c_{p^+(V_i)})$





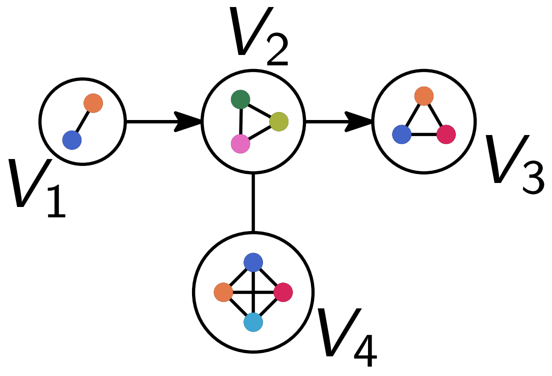
FPT Mixed Neighborhood Diversity

Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

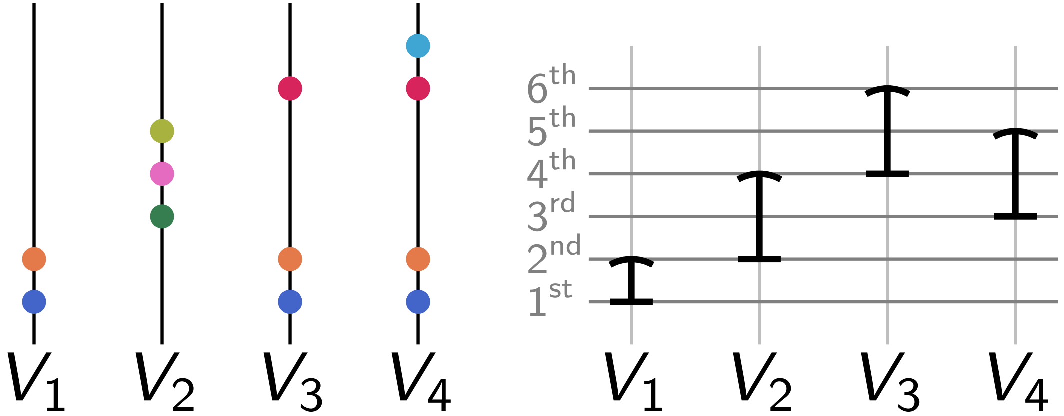
Lemma

coloring c respects arcs if $\forall (V_i, V_j) \in A \colon c^+(V_i) \leq c^-(V_j)$



Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t
- proper if $\forall (V_i, V_j) \in A \colon p^+(V_i) \leq p^-(V_j)$
- coloring c corresponds to preorder p if $\exists c_1 < \cdots < c_\ell \colon c(V_i) \subset [c_{p^-(V_i)}, c_{p^+(V_i)}]$



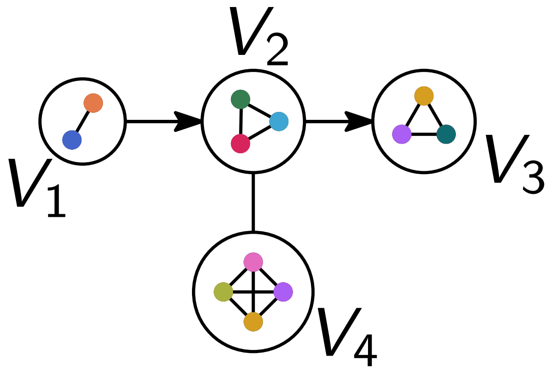
FPT Mixed Neighborhood Diversity

Theorem

$\textsc{MixedColoring}$ is FPT w.r.t. mixed neighborhood diversity

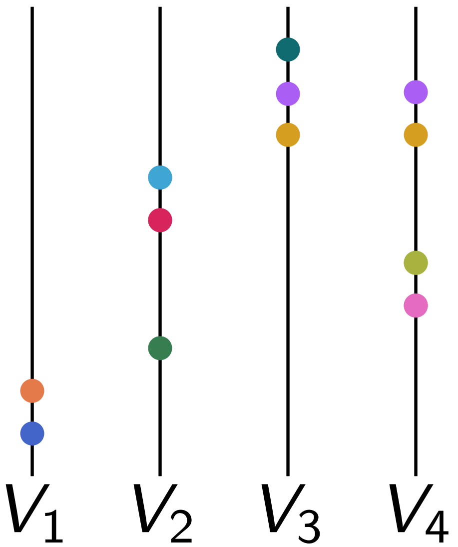
Lemma

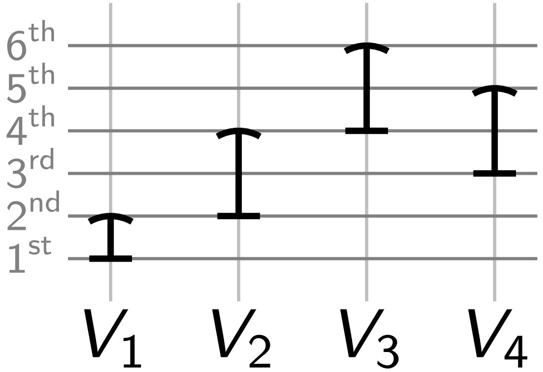
coloring c respects arcs if $\forall (V_i, V_j) \in A \colon c^+(V_i) \leq c^-(V_j)$



Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t
- proper if $\forall (V_i, V_j) \in A \colon p^+(V_i) \leq p^-(V_j)$
- coloring c corresponds to preorder p if $\exists c_1 < \cdots < c_\ell \colon c(V_i) \subset [c_{p^-(V_i)}, c_{p^+(V_i)})$





FPT Mixed Neighborhood Diversity

Theorem

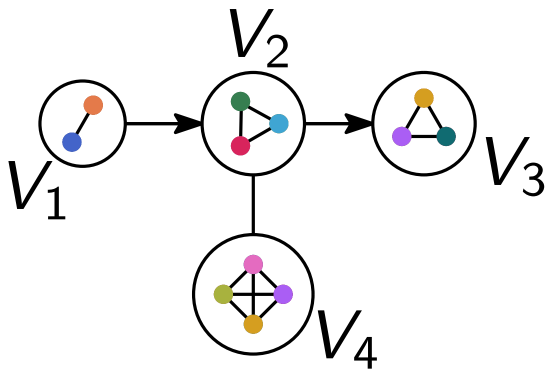
MixedColoring is FPT w.r.t. mixed neighborhood diversity

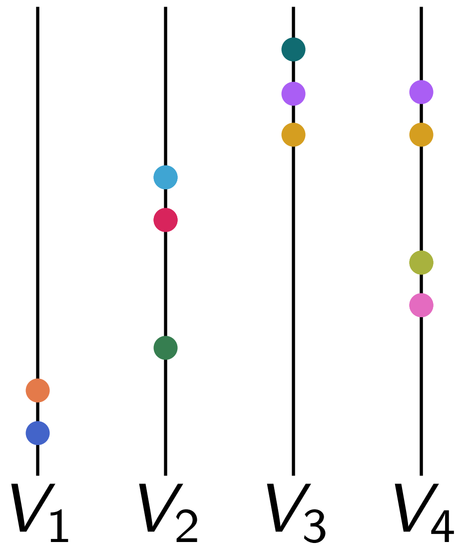
Lemma

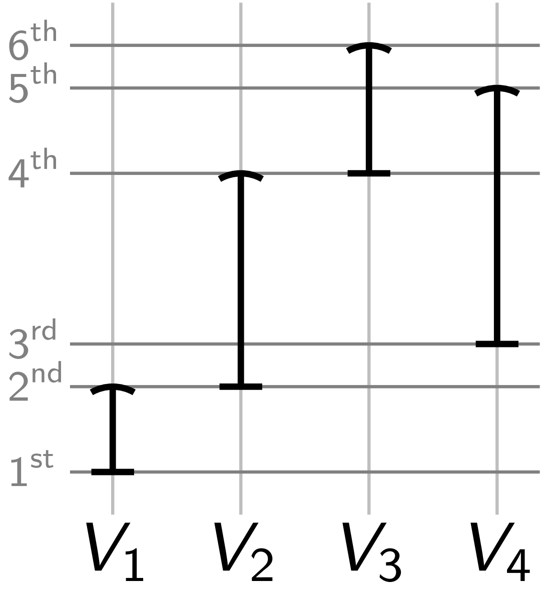
coloring c respects arcs if $\forall (V_i, V_j) \in A\colon c^+(V_i) \leq c^-(V_j)$

Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t
- proper if $\forall (V_i, V_j) \in A\colon p^+(V_i) \leq p^-(V_j)$
- coloring c corresponds to preorder p if $\exists c_1 < \cdots < c_\ell\colon c(V_i) \subset [c_{p^-(V_i)}, c_{p^+(V_i)})$







FPT Mixed Neighborhood Diversity

Theorem

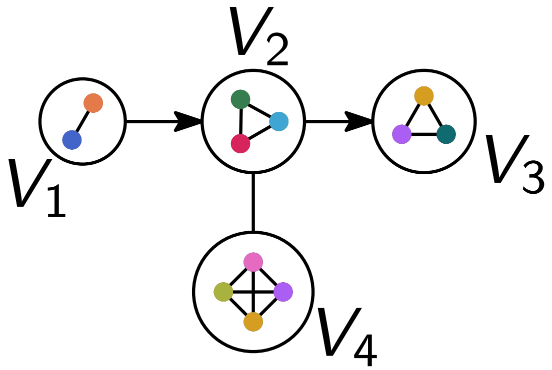
MixedColoring is FPT w.r.t. mixed neighborhood diversity

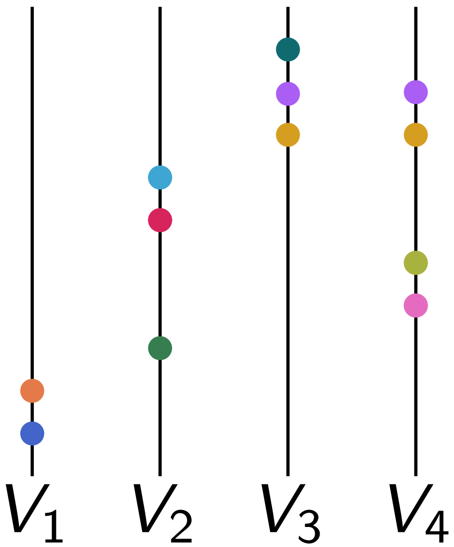
Lemma

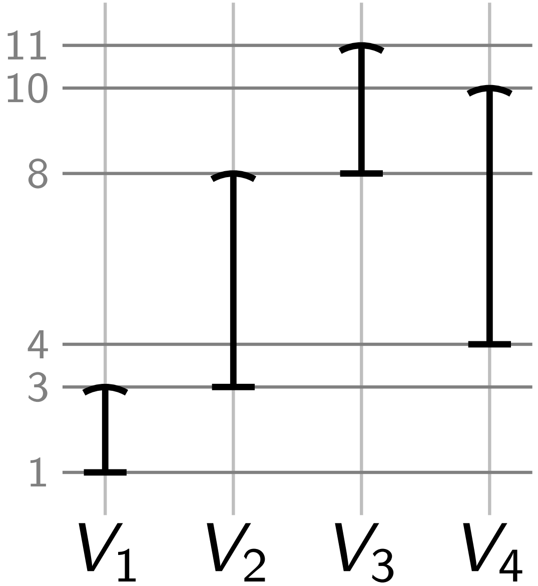
coloring c respects arcs if $\forall (V_i, V_j) \in A\colon c^+(V_i) \leq c^-(V_j)$

Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t
- proper if $\forall (V_i, V_j) \in A\colon p^+(V_i) \leq p^-(V_j)$
- coloring c corresponds to preorder p if $\exists c_1 < \cdots < c_\ell\colon c(V_i) \subset [c_{p^-(V_i)}, c_{p^+(V_i)})$







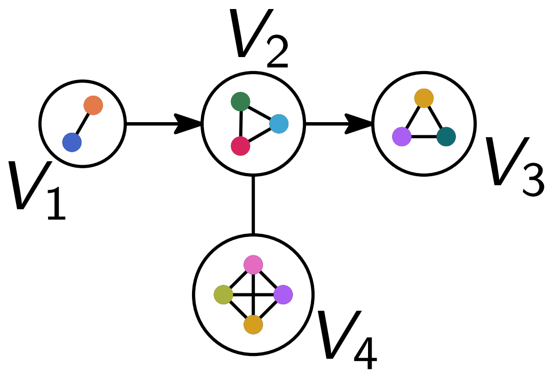
FPT Mixed Neighborhood Diversity

Theorem

$\textsc{MixedColoring}$ is FPT w.r.t. mixed neighborhood diversity

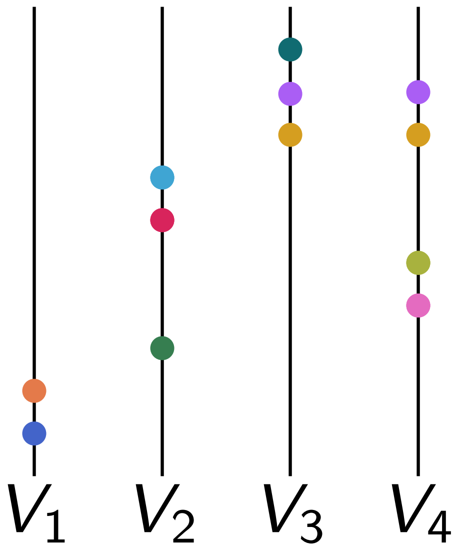
Lemma

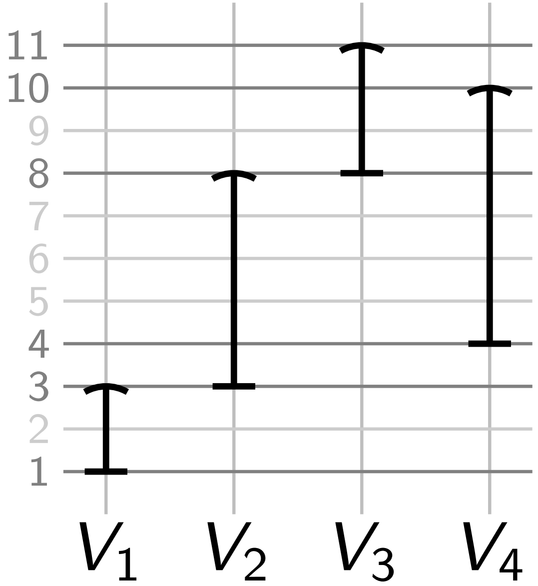
coloring c respects arcs if $\forall (V_i, V_j) \in A\colon c^+(V_i) \leq c^-(V_j)$



Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t
- proper if $\forall (V_i, V_j) \in A\colon p^+(V_i) \leq p^-(V_j)$
- coloring c corresponds to preorder p if $\exists c_1 < \cdots < c_\ell\colon c(V_i) \subset [c_{p^-(V_i)}, c_{p^+(V_i)})$

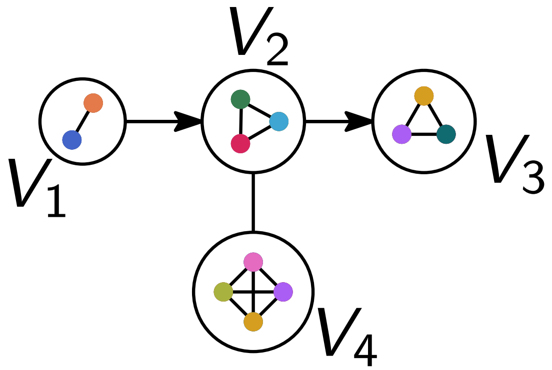




FPT Mixed Neighborhood Diversity

Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

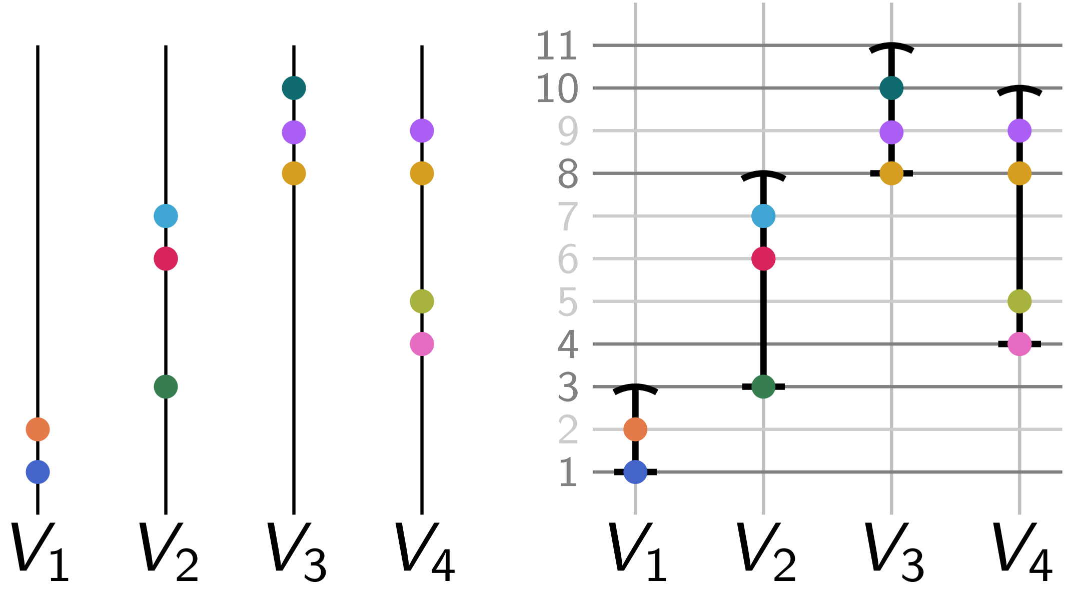


Lemma

coloring c respects arcs if $\forall (V_i, V_j) \in A \colon c^+(V_i) \leq c^-(V_j)$

Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t
- proper if $\forall (V_i, V_j) \in A \colon p^+(V_i) \leq p^-(V_j)$
- coloring c corresponds to preorder p if $\exists c_1 < \cdots < c_\ell \colon c(V_i) \subset [c_{p^-(V_i)}, c_{p^+(V_i)})$



FPT Mixed Neighborhood Diversity

Theorem

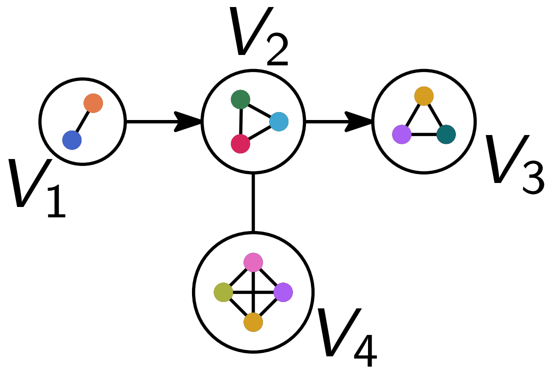
MixedColoring is FPT w.r.t. mixed neighborhood diversity

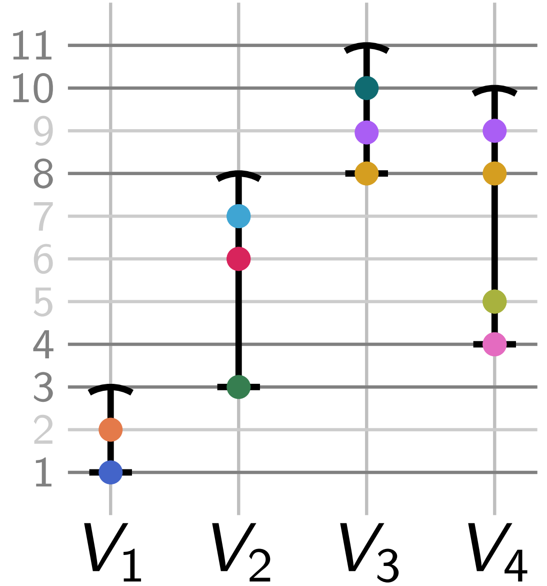
Lemma

coloring c respects arcs if $\forall(V_i, V_j) \in A\colon c^+(V_i) \le c^-(V_j)$

Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t
- proper if $\forall(V_i, V_j) \in A\colon p^+(V_i) \le p^-(V_j)$
- coloring c corresponds to preorder p if $\exists c_1 < \cdots < c_\ell\colon c(V_i) \subset [c_{p^-(V_i)}, c_{p^+(V_i)})$





FPT Mixed Neighborhood Diversity

Theorem
MixedColoring is FPT w.r.t. mixed neighborhood diversity

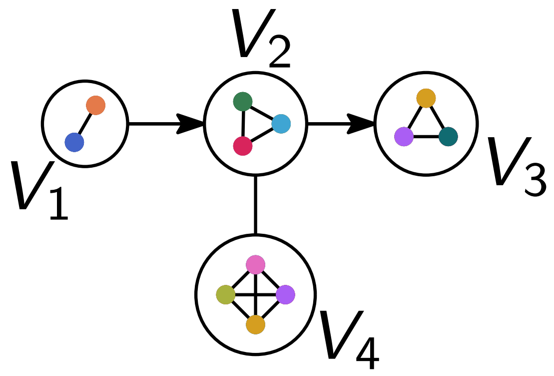
Lemma
coloring c respects arcs if $\forall (V_i, V_j) \in A\colon c^+(V_i) \leq c^-(V_j)$

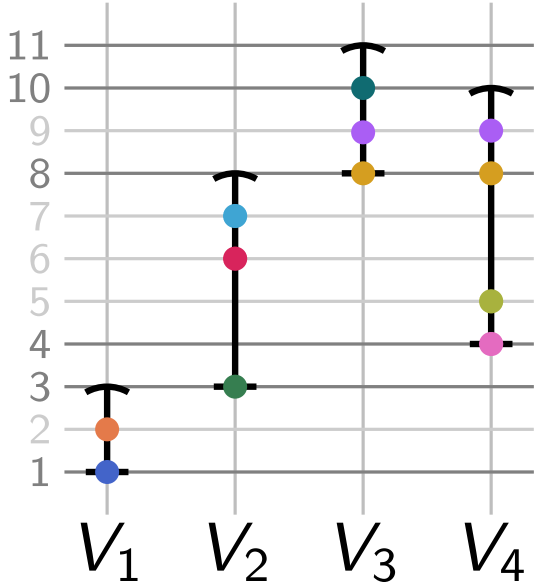
Type Preorder

- assigns $p^-(V_t) < p^+(V_t)$ for each type t
- proper if $\forall (V_i, V_j) \in A\colon p^+(V_i) \leq p^-(V_j)$
- coloring c corresponds to preorder p if $\exists c_1 < \cdots < c_\ell\colon c(V_i) \subset [c_{p^-(V_i)}, c_{p^+(V_i)})$

Lemma

- coloring corresponding to proper preorder respects arcs
- each proper coloring corresponds to at least one proper preorder





FPT Mixed Neighborhood Diversity

Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

Algorithm

FPT Mixed Neighborhood Diversity

Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

Algorithm

```
foreach proper preorder do
    check if there exists a corresponding proper $k$-coloring
```

FPT Mixed Neighborhood Diversity

Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

Algorithm

$2^{\mathcal{O}(\text{mnd} \log \text{mnd})}$

```
foreach proper preorder do
    check if there exists a corresponding proper $k$-coloring
```

FPT Mixed Neighborhood Diversity

Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

Algorithm

$2^{\mathcal{O}(\text{mnd} \log \text{mnd})}$

```
foreach proper preorder do
    check if there exists a corresponding proper $k$-coloring
```

Lemma

finding a proper k-coloring corresponding to a proper preorder is FPT w.r.t. mixed neighborhood diversity

FPT Mixed Neighborhood Diversity

Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

Algorithm

$2^{\mathcal{O}(\text{mnd log mnd})}$

```
foreach proper preorder do
    check if there exists a corresponding proper $k$-coloring
```

$\implies$ FPT w.r.t. mixed neighborhood diversity

Lemma

finding a proper k-coloring corresponding to a proper preorder is FPT w.r.t. mixed neighborhood diversity

FPT Mixed Neighborhood Diversity

Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

Algorithm

$2^{\mathcal{O}(\text{mnd}\log\text{mnd})}$

```
foreach proper preorder do
    check if there exists a corresponding proper $k$-coloring
```

$\implies$ FPT w.r.t. mixed neighborhood diversity

Lemma

finding a proper k-coloring corresponding to a proper preorder is FPT w.r.t. mixed neighborhood diversity

Proof (Idea):

FPT Mixed Neighborhood Diversity

Theorem
MixedColoring is FPT w.r.t. mixed neighborhood diversity

Algorithm

$2^{\mathcal{O}(\text{mnd} \log \text{mnd})}$

```
foreach proper preorder do
    check if there exists a corresponding proper $k$-coloring
```

$\implies$ FPT w.r.t. mixed neighborhood diversity

Lemma
finding a proper k-coloring corresponding to a proper preorder is FPT w.r.t. mixed neighborhood diversity

Proof (Idea):

- arcs no longer relevant

FPT Mixed Neighborhood Diversity

Theorem

MixedColoring is FPT w.r.t. mixed neighborhood diversity

Algorithm

$2^{\mathcal{O}(\text{mnd} \log \text{mnd})}$

```
foreach proper preorder do
    check if there exists a corresponding proper $k$-coloring
```

$\Longrightarrow$ FPT w.r.t. mixed neighborhood diversity

Lemma

finding a proper *k*-coloring corresponding to a proper preorder is FPT w.r.t. mixed neighborhood diversity

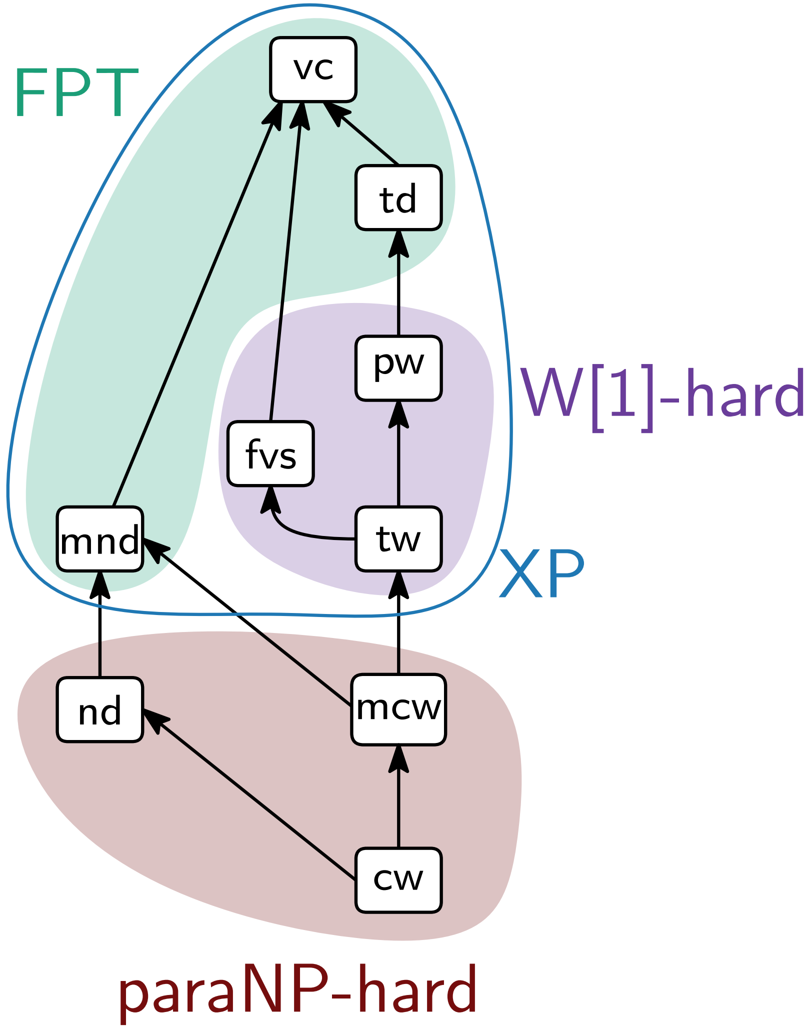
Proof (Idea):

- arcs no longer relevant
- similar approach as FPT-Algo w.r.t. neighborhood diversity for Coloring

Conclusion

Complexity

MixedColoring

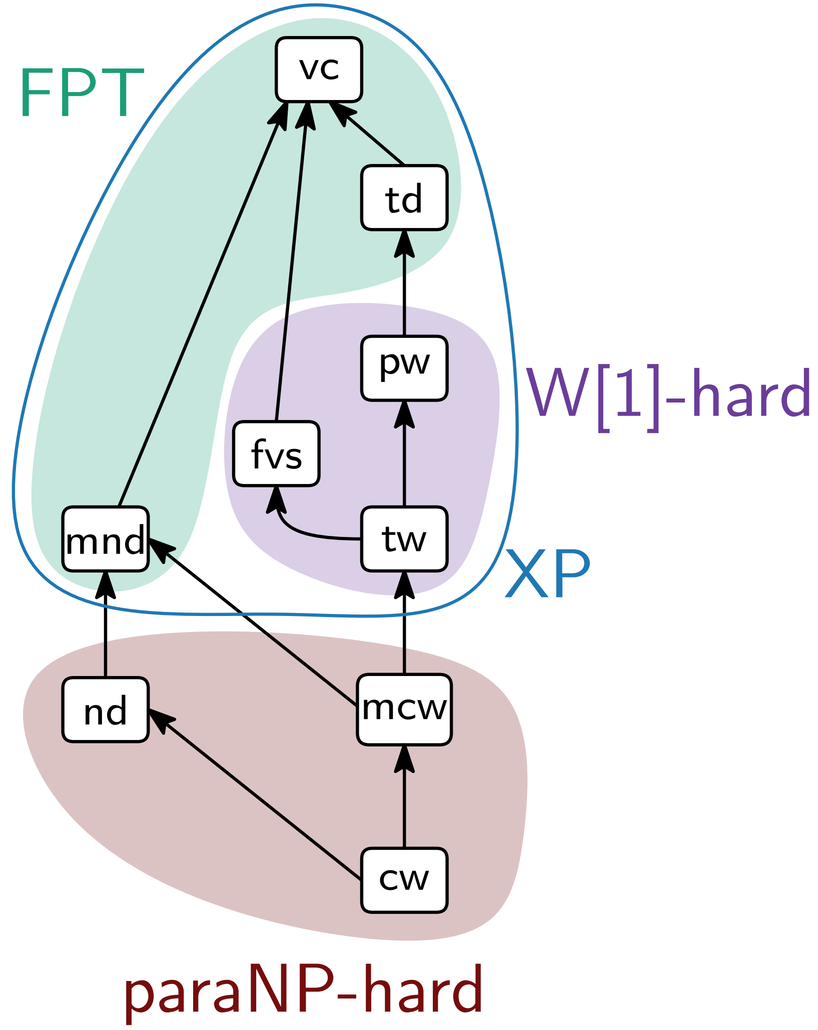


Conclusion

Open Problems

Complexity

MixedColoring



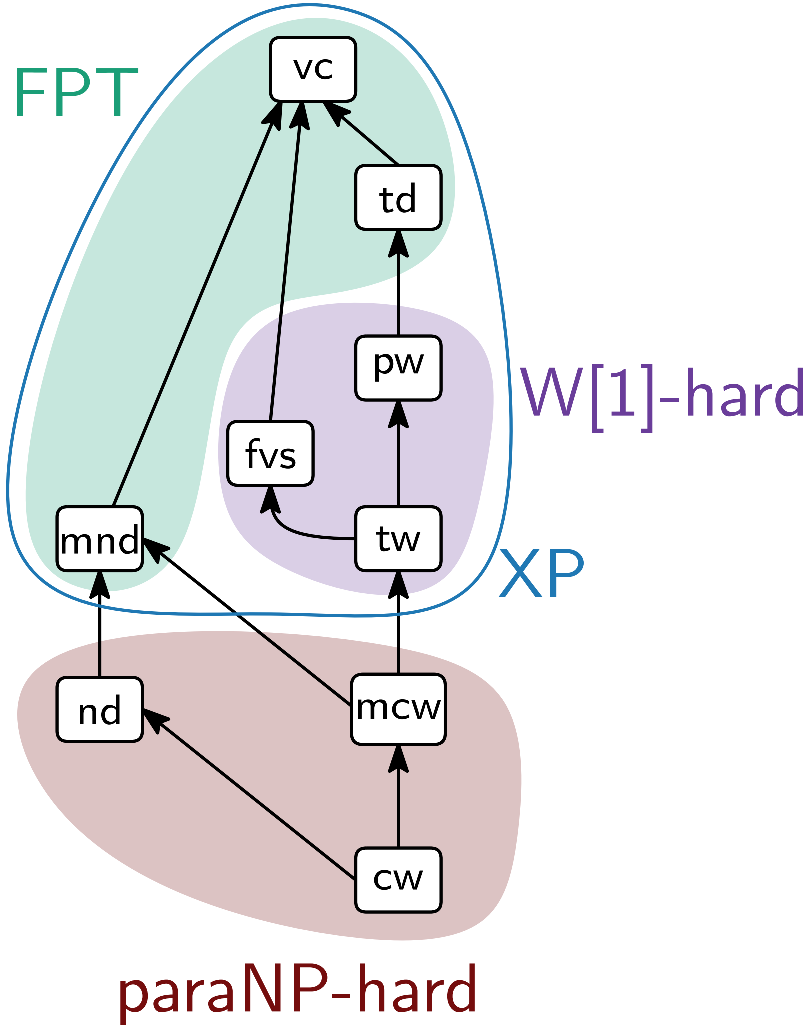
Conclusion

Open Problems

Further Parameters

Complexity

MixedColoring



Conclusion

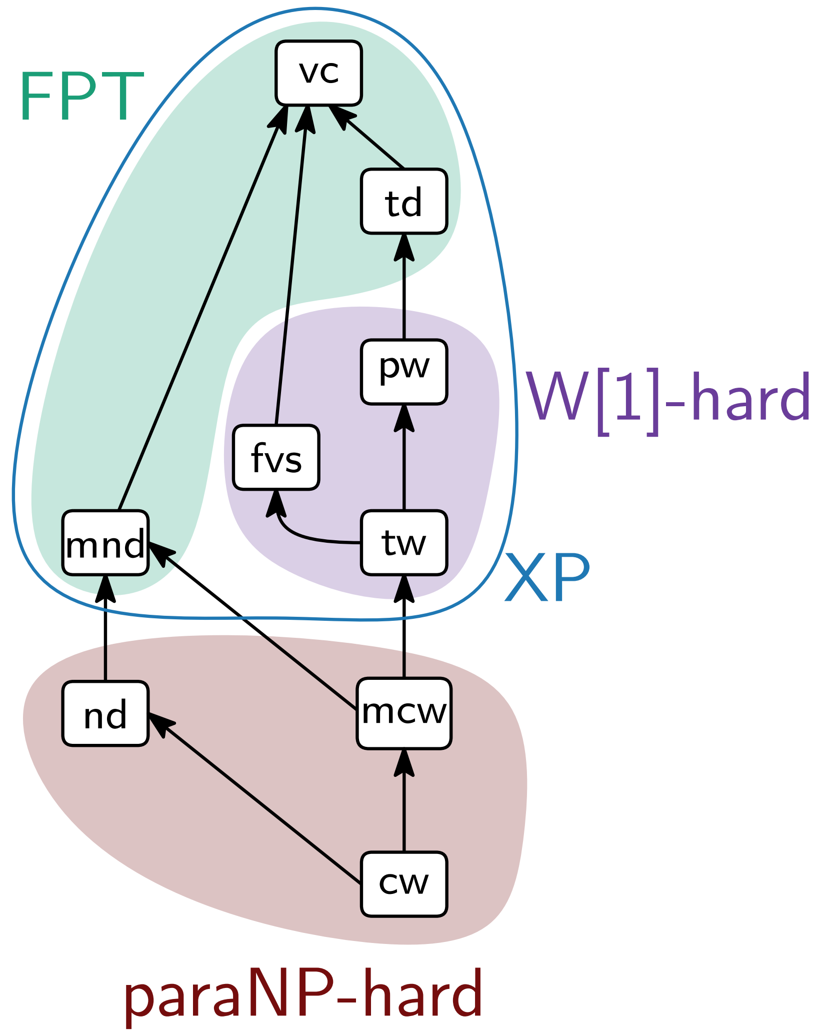
Open Problems

Further Parameters

- mixed variants

Complexity

MixedColoring



Conclusion

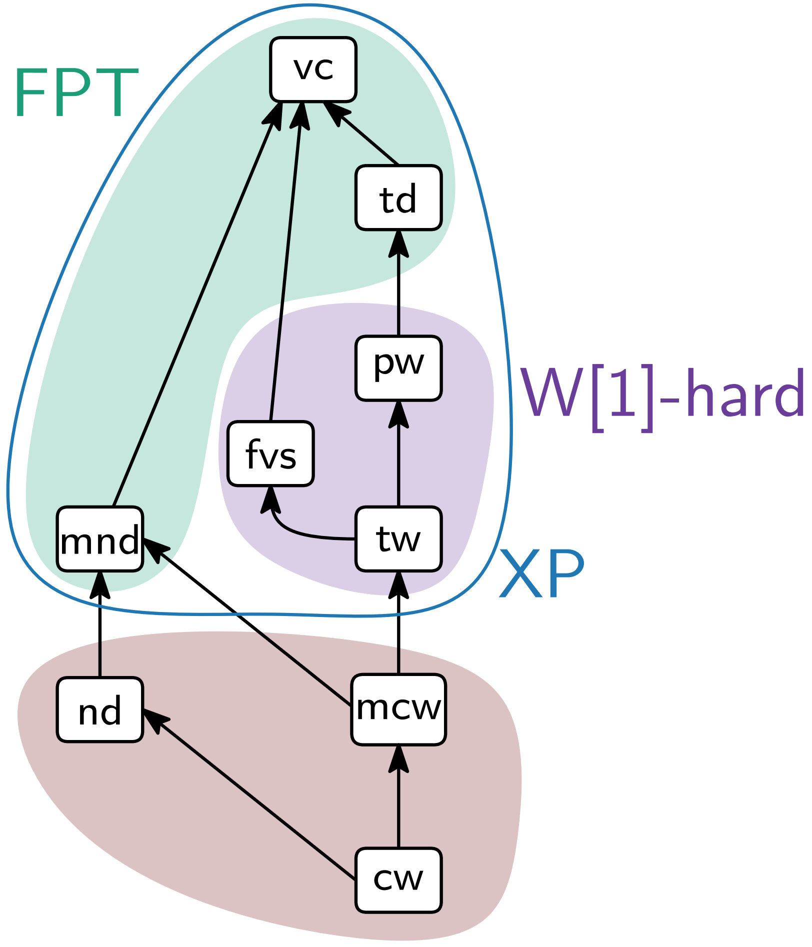
Open Problems

Further Parameters

- mixed variants
- complexity

Complexity

MixedColoring



paraNP-hard

Conclusion

Open Problems

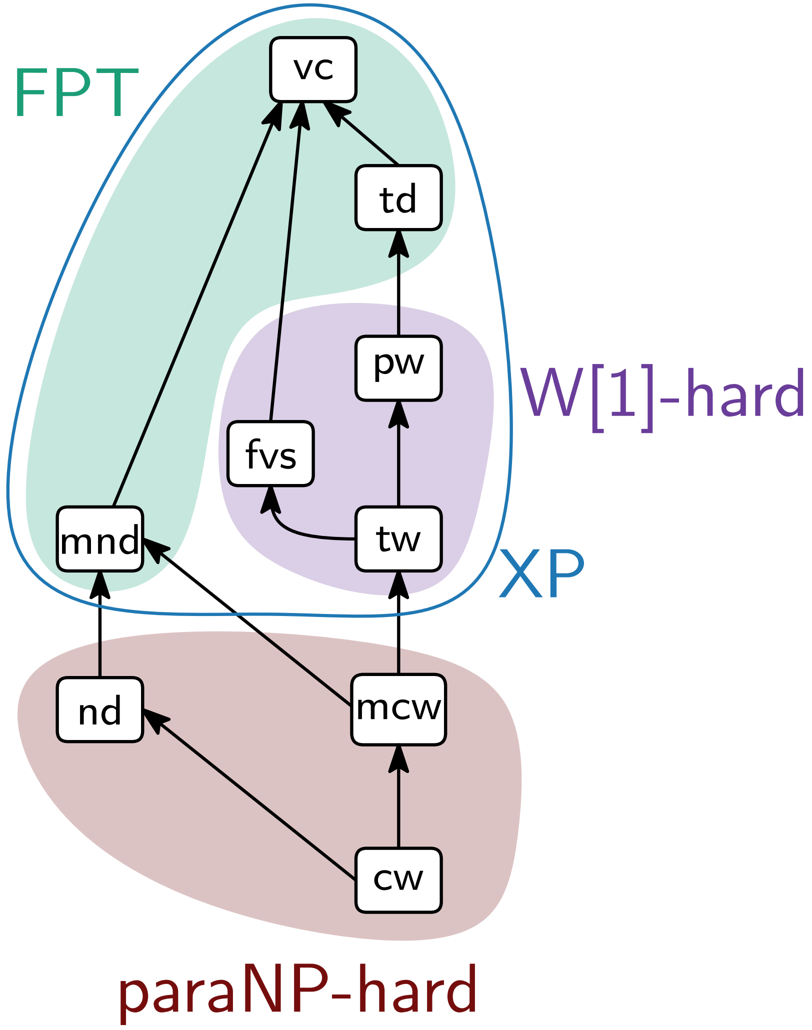
Further Parameters

- mixed variants
- complexity

Meta Theorems

Complexity

Mixed Coloring



Conclusion

Open Problems

Further Parameters

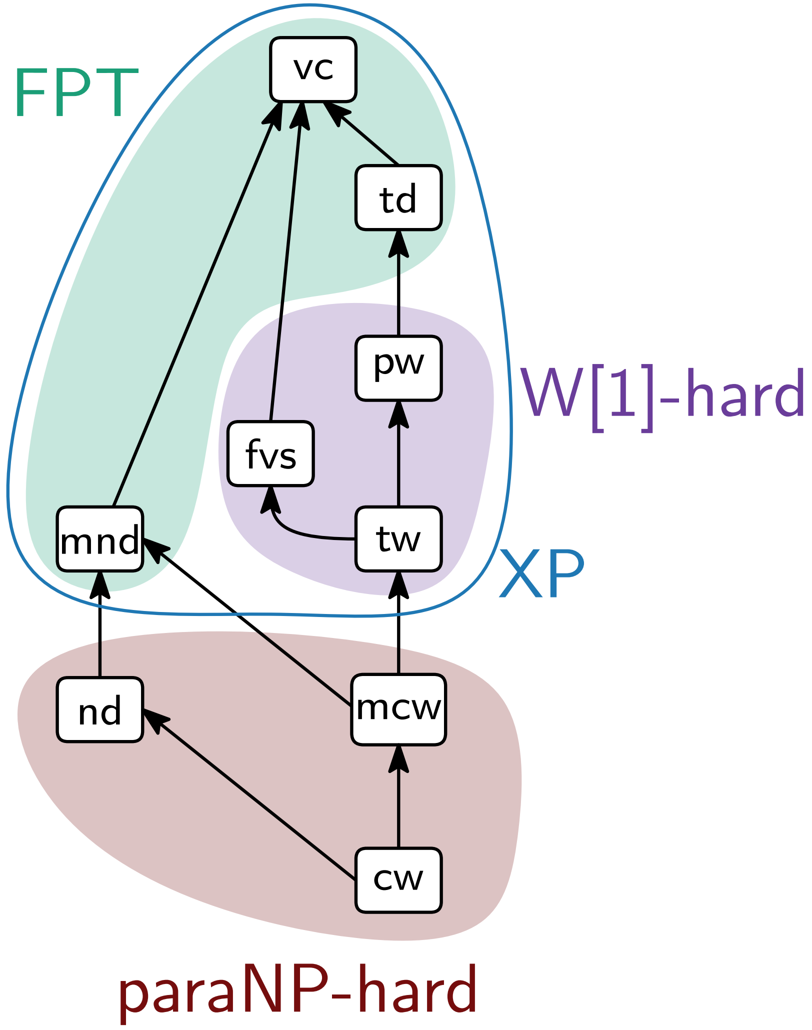
- mixed variants
- complexity

Meta Theorems

- Courcelle's Theorem

Complexity

MixedColoring



Conclusion

Open Problems

Further Parameters

- mixed variants
- complexity

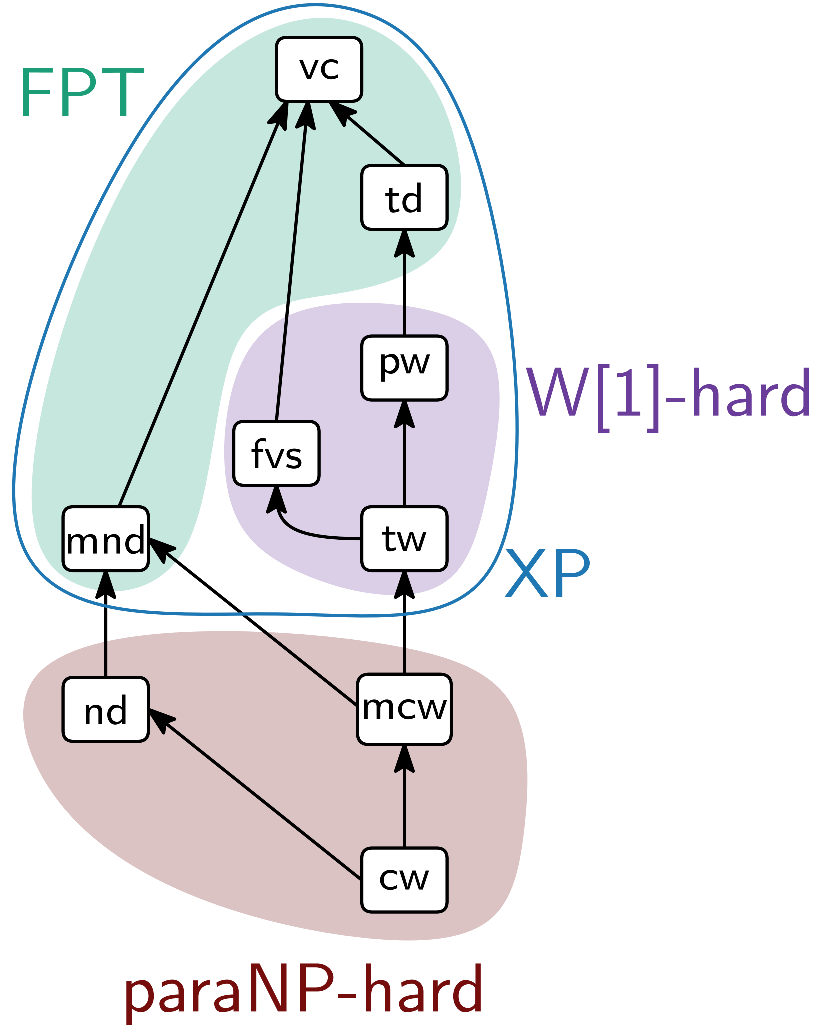
Meta Theorems

- Courcelle's Theorem

Variants of Coloring

Complexity

MixedColoring



Conclusion

Open Problems

Further Parameters

- mixed variants
- complexity

Meta Theorems

- Courcelle's Theorem

Variants of Coloring

- ListColoring

Complexity

MixedColoring