Reconstructing perfect phylogenies via binary matrices, branchings in DAGs, and a generalization of Dilworth's theorem

Martin Milanič

IAM and FAMNIT, University of Primorska, Koper, Slovenia

LAGOS IX, CIRM, Marseille, September 14, 2017

Urša Kačar, Celtra Inc.

Bernard Ries, University of Fribourg

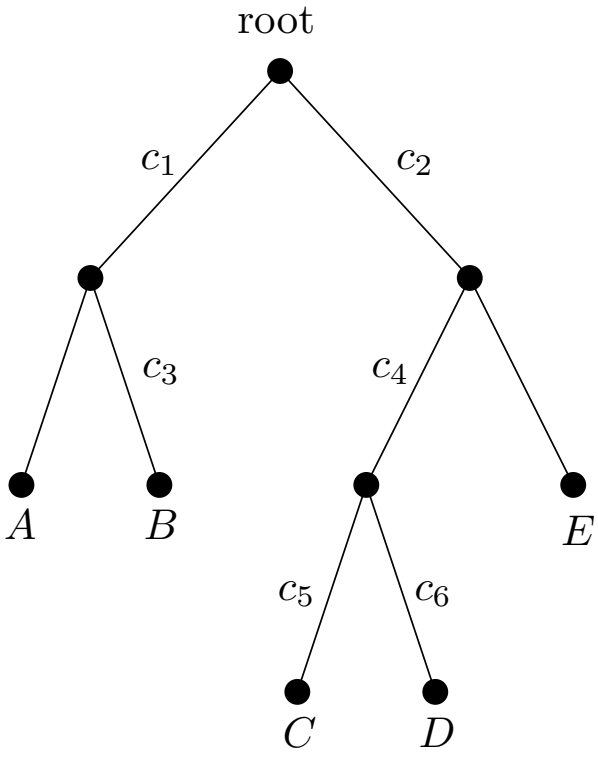
Ademir Hujdurović, University of Primorska

Alexandru I. Tomescu, University of Helsinki

Romeo Rizzi, University of Verona

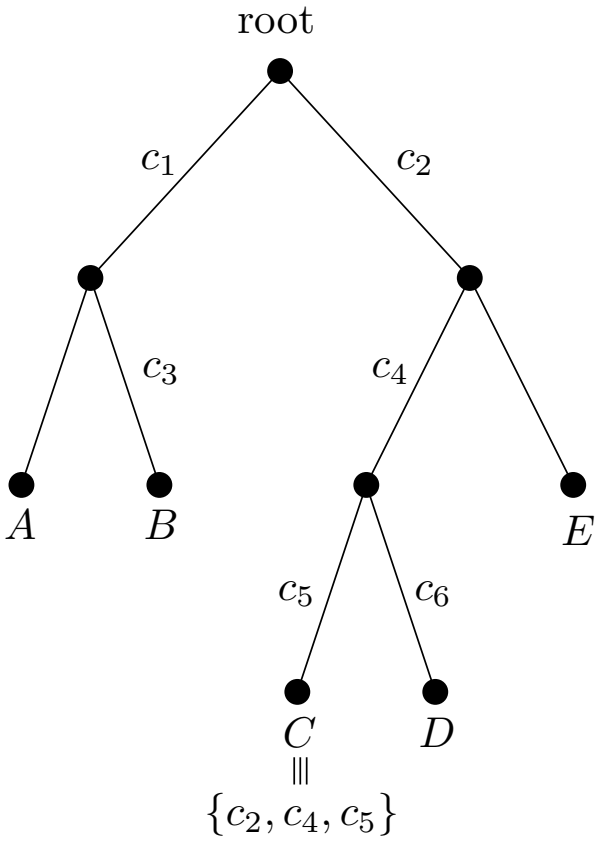
Edin Husić, ENS Lyon and University of Primorska (soon at LSE)

The problems



A **perfect phylogeny** is a rooted tree representing the evolutionary history of a set of m objects such that:

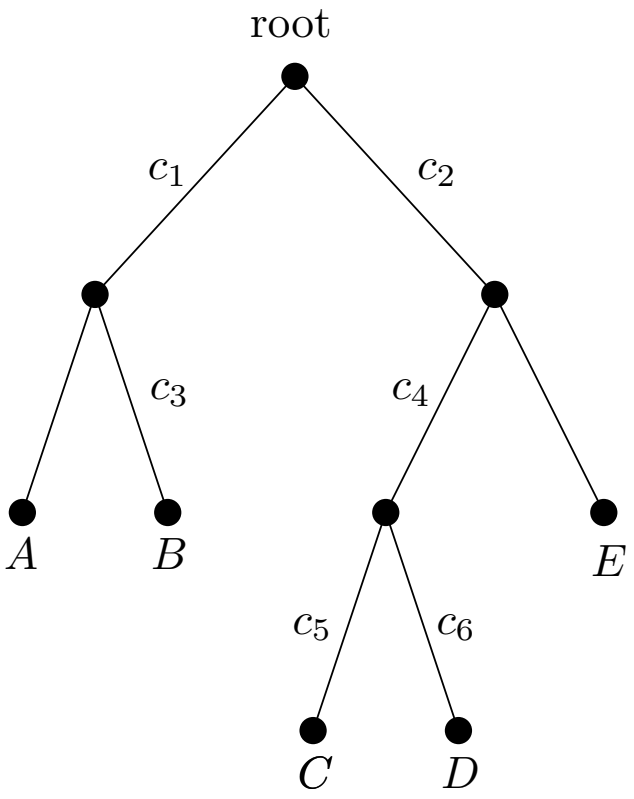
- The objects bijectively label the leaves of the tree.
- There are n **binary characters**, each labeling exactly one edge of the tree.
- For each leaf, the set of characters that appear on the unique root-to-leaf path is the set of characters taking value 1 at the object labeling the leaf.



A **perfect phylogeny** is a rooted tree representing the evolutionary history of a set of m objects such that:

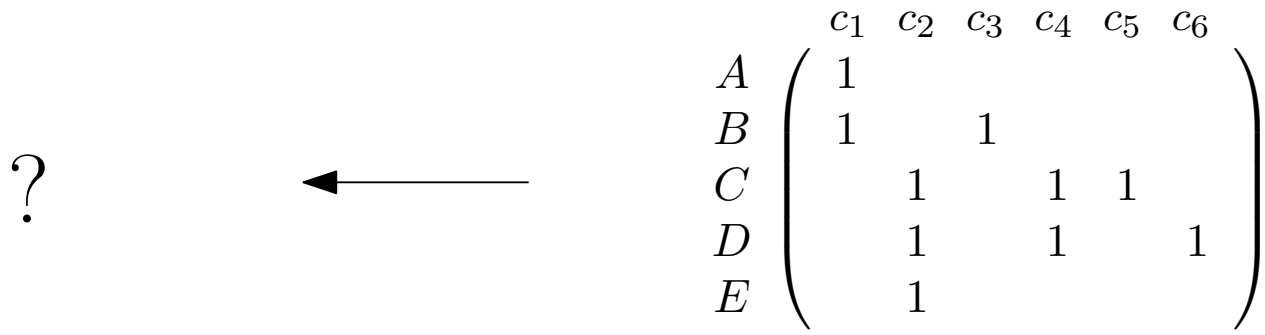
- The objects bijectively label the leaves of the tree.
- There are n **binary characters**, each labeling exactly one edge of the tree.
- For each leaf, the set of characters that appear on the unique root-to-leaf path is the set of characters taking value 1 at the object labeling the leaf.

Every perfect phylogeny naturally corresponds to an $m \times n$ **binary matrix** having objects as rows and characters as columns:



$$
\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array}
\begin{array}{c}
\begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array} \\
\left(\begin{array}{cccccc}
1 & & & & & \\
1 & & 1 & & & \\
 & 1 & & 1 & 1 & \\
 & 1 & & 1 & & 1 \\
 & 1 & & & &
\end{array} \right)
\end{array}
$$

The **perfect phylogeny problem** asks the opposite question:
Does a given binary matrix correspond to a perfect phylogeny?



The perfect phylogeny problem is solved:

a binary matrix corresponds to a perfect phylogeny if and only if it is conflict-free. [Estabrook et al. 1975, Gusfield 1991].

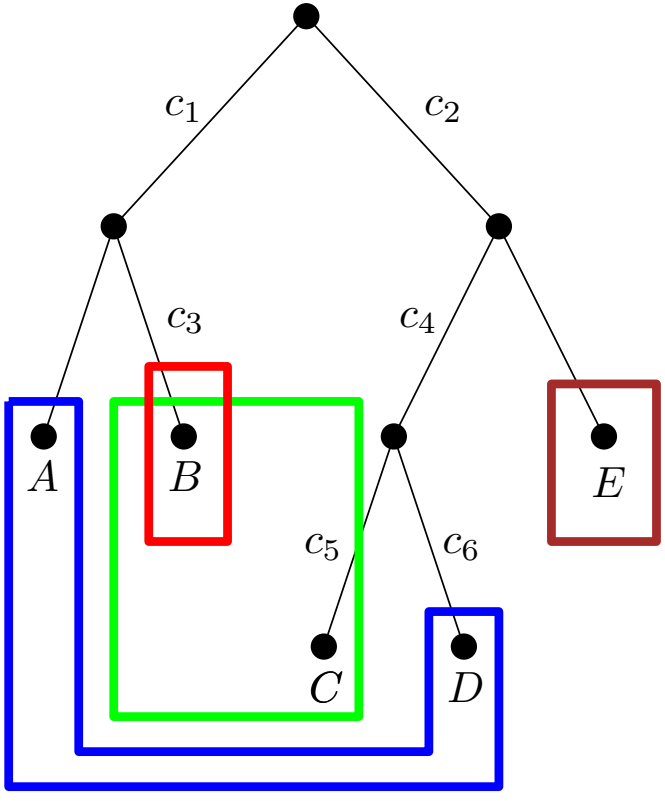
Two columns i and j of a binary matrix M are said to be **in conflict** if there exist rows r, r', r'' of M such that

$$M[(r, r', r''), (i, j)] = \begin{pmatrix} 1 & 1 \\ 1 & \\ & 1 \end{pmatrix} .$$

A binary matrix M is **conflict-free** if no two columns of M are in conflict.

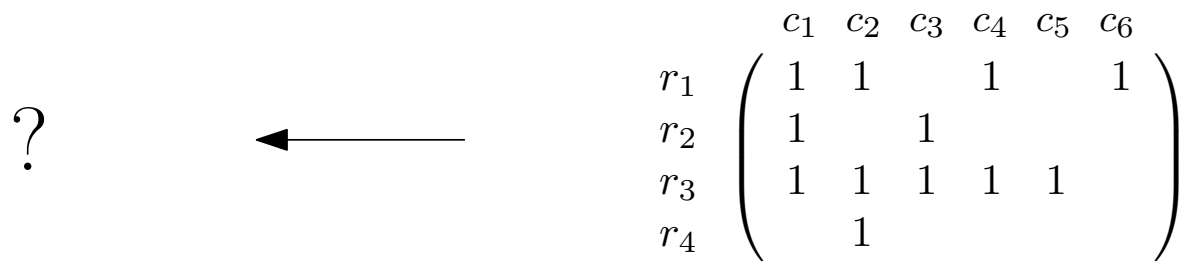
The perfect phylogeny problem and various generalizations of it have been extensively studied in computational biology.

We look at **two generalizations**, first considered by
Hajirasouliha and Raphael (WABI 2014)
and motivated by applications in cancer genomics.



$$
\begin{array}{r} \\ \{A,D\} \\ \{B\} \\ \{B,C\} \\ \{E\} \end{array}
\begin{array}{c}
\begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array} \\
\left(\begin{array}{cccccc}
1 & 1 & & 1 & & 1 \\
1 & & 1 & & & \\
1 & 1 & 1 & 1 & 1 & \\
 & 1 & & & &
\end{array} \right)
\end{array}
$$

We look at **two generalizations**, first considered by
Hajirasouliha and Raphael (WABI 2014)
and motivated by applications in cancer genomics.



How can a given binary matrix be explained in the simplest possible way?

This question leads to two optimization problems.

1. The minimum conflict-free row split (MCRS) problem:

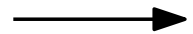
Split each row of a given binary matrix into a bitwise OR of a set of rows so that the resulting matrix is conflict-free

(that is, it corresponds to a perfect phylogeny)

and has the **minimum number of rows** *among all matrices with this property.*

M

	c_1	c_2	c_3	c_4	c_5	c_6
r_1	1	1		1		1
r_2	1		1			
r_3	1	1	1	1	1	
r_4		1				



M'

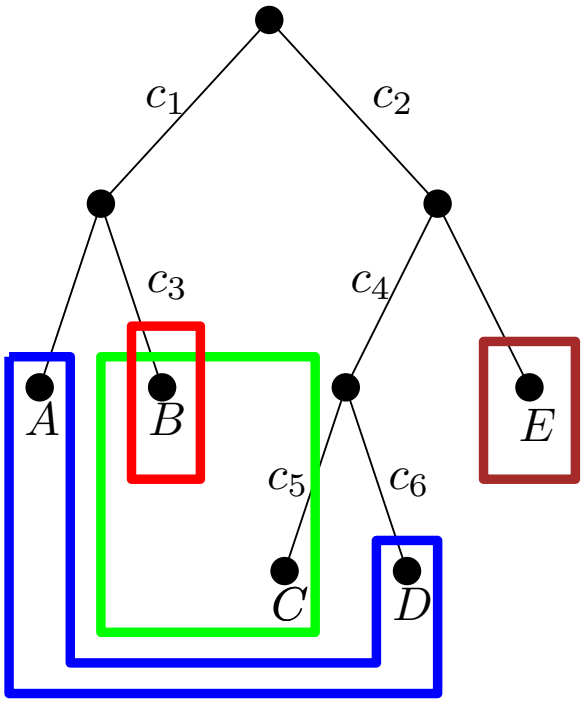
	c_1	c_2	c_3	c_4	c_5	c_6
R_1	1					
		1		1		1
R_2	1		1			
R_3	1		1			
		1		1	1	
R_4		1				

1. The minimum conflict-free row split (MCRS) problem:

Split each row of a given binary matrix into a bitwise OR of a set of rows so that the resulting matrix is conflict-free

(that is, it corresponds to a perfect phylogeny)

and has the **minimum number of rows** *among all matrices with this property.*



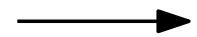
$$M' = \begin{array}{c} \\ R_1 \\ R_2 \\ R_3 \\ R_4 \end{array} \begin{array}{c} \begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array} \\ \left(\begin{array}{cccccc} 1 & & & & & \\ & 1 & & 1 & & 1 \\ \hline 1 & & 1 & & & \\ \hline 1 & & 1 & & & \\ & 1 & & 1 & 1 & \\ \hline & 1 & & & & \end{array} \right) \end{array}$$

2. The minimum distinct conflict-free row split (MDCRS) problem:

the variant of the problem in which the task is to minimize the number of **distinct** *rows of the resulting matrix.*

M

	c_1	c_2	c_3	c_4	c_5	c_6
r_1	1	1		1		1
r_2	1		1			
r_3	1	1	1	1	1	
r_4		1				



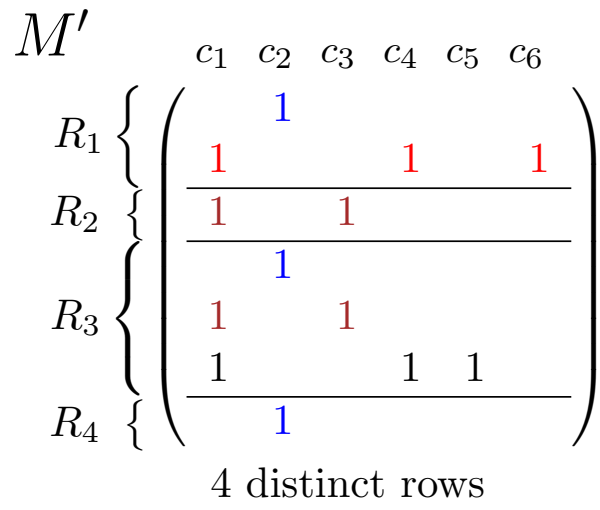
M'

	c_1	c_2	c_3	c_4	c_5	c_6
R_1	1					
		1		1		1
R_2	1		1			
R_3	1		1			
		1		1	1	
R_4		1				

5 distinct rows

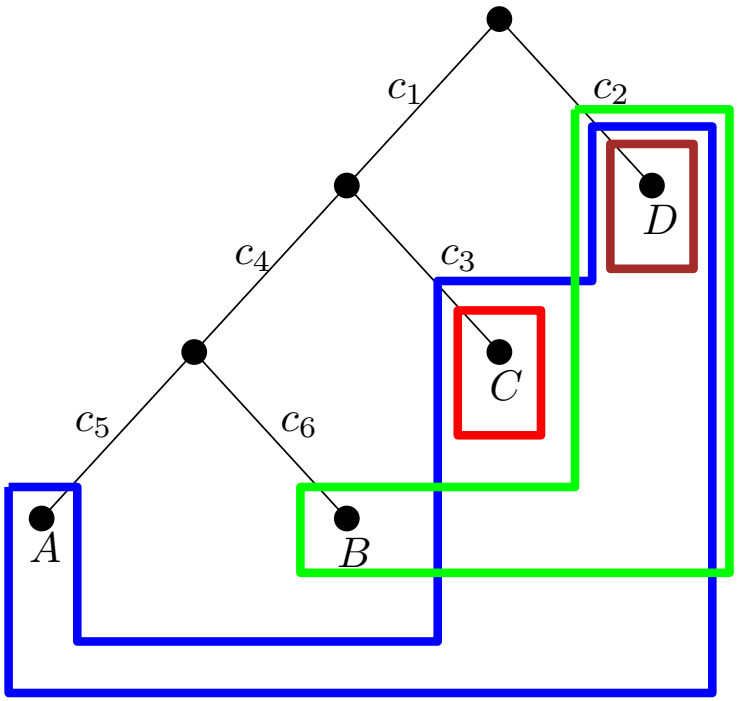
2. The minimum distinct conflict-free row split (MDCRS) problem:

the variant of the problem in which the task is to minimize the number of **distinct** *rows of the resulting matrix.*



2. The minimum distinct conflict-free row split (MDCRS) problem:

the variant of the problem in which the task is to minimize the number of **distinct** *rows of the resulting matrix.*



$$M' = \begin{array}{c} \\ R_1 \\ R_2 \\ R_3 \\ R_4 \end{array} \begin{array}{c} \begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array} \\ \left(\begin{array}{cccccc} & 1 & & & & \\ 1 & & & 1 & & 1 \\ \hline 1 & & 1 & & & \\ \hline & 1 & & & & \\ 1 & & 1 & & & \\ 1 & & & 1 & 1 & \\ \hline & 1 & & & & \end{array} \right) \end{array}$$

4 distinct rows

Previous result: a chromatic lower bound

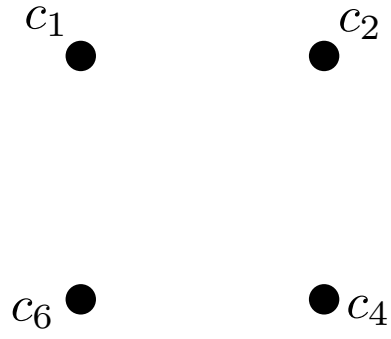
Given a binary matrix M and a row r of M, let us denote by $G_{M,r}$ the so-called **row-conflict graph**, where:

- we associate a vertex to each entry 1 in r,
- edges represent conflicting column pairs.

M

$$
\begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{array}
\begin{array}{c}
\begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array} \\
\left(\begin{array}{cccccc}
\mathbf{1} & \mathbf{1} & & \mathbf{1} & & \mathbf{1} \\
1 & & 1 & & & \\
1 & 1 & 1 & 1 & 1 & \\
 & 1 & & & &
\end{array} \right)
\end{array}
$$

$G_{M,\mathbf{r_1}}$

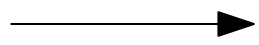


Given a binary matrix M and a row r of M, let us denote by $G_{M,r}$ the so-called **row-conflict graph**, where:

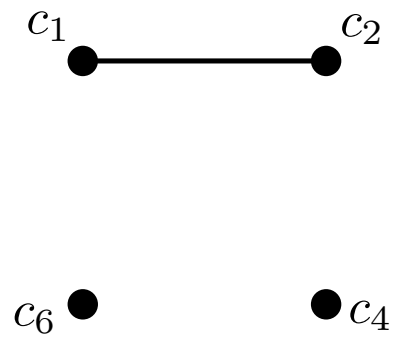
- we associate a vertex to each entry 1 in r,
- edges represent conflicting column pairs.

M

	c_1	c_2	c_3	c_4	c_5	c_6
r_1	**1**	**1**		**1**		**1**
r_2	**1**	**0**	1			
r_3	**1**	**1**	1	1	1	
r_4	**0**	**1**				



$G_{M,\mathbf{r_1}}$



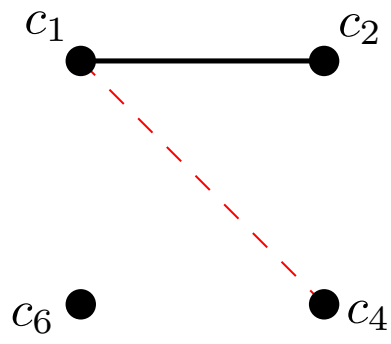
Given a binary matrix M and a row r of M, let us denote by $G_{M,r}$ the so-called **row-conflict graph**, where:

- we associate a vertex to each entry 1 in r,
- edges represent conflicting column pairs.

M

$$
\begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{array}
\begin{array}{c}
\begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array} \\
\left(\begin{array}{cccccc}
\mathbf{1} & \mathbf{1} & & \mathbf{1} & & \mathbf{1} \\
\mathbf{1} & & 1 & & & \\
\mathbf{1} & 1 & 1 & \mathbf{1} & 1 & \\
 & 1 & & & &
\end{array} \right)
\end{array}
$$

$G_{M,\mathbf{r_1}}$



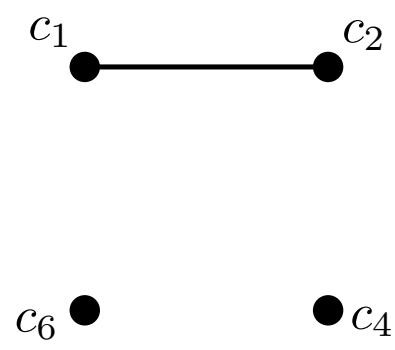
Given a binary matrix M and a row r of M, let us denote by $G_{M,r}$ the so-called **row-conflict graph**, where:

- we associate a vertex to each entry 1 in r,
- edges represent conflicting column pairs.

M

$$
\begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{array}
\begin{array}{c}
\begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array} \\
\begin{pmatrix}
\mathbf{1} & \mathbf{1} & & \mathbf{1} & & \mathbf{1} \\
1 & & 1 & & & \\
1 & 1 & 1 & 1 & 1 & \\
 & 1 & & & &
\end{pmatrix}
\end{array}
$$

$G_{M,\mathbf{r_1}}$



In the paper introducing the problem,
Hajirasouliha and Raphael (WABI 2014)
proved that in order to resolve conflicts, each row of M needs to be split into at least $\chi(G_{M,r})$ rows.

Hence,

$$\mathrm{OPT}_{\mathrm{MCRS}}(M) \geq \sum_{r} \chi(G_{M,r})\,.$$

Part 1: results from ...

Hujdurović-Kačar-M-Ries-Tomescu (IEEE TCBB, 2016)

Urša Kačar, Celtra Inc.

Bernard Ries, University of Fribourg

Ademir Hujdurović, University of Primorska

Alexandru I. Tomescu, University of Helsinki

A characterization of row-conflict graphs

Hajirasouliha and Raphael claimed that every graph G is a row-conflict graph.

Well, not really . . .

We show:

Proposition

A graph G is a row-conflict graph if and only if its complement is **transitively orientable***.*

Transitively orientable graphs are also called comparability graphs.

$\Rightarrow$ row-conflict graphs are exactly the **cocomparability graphs**.

The list of forbidden induced subgraphs for cocomparability graphs is known (Gallai, 1967):

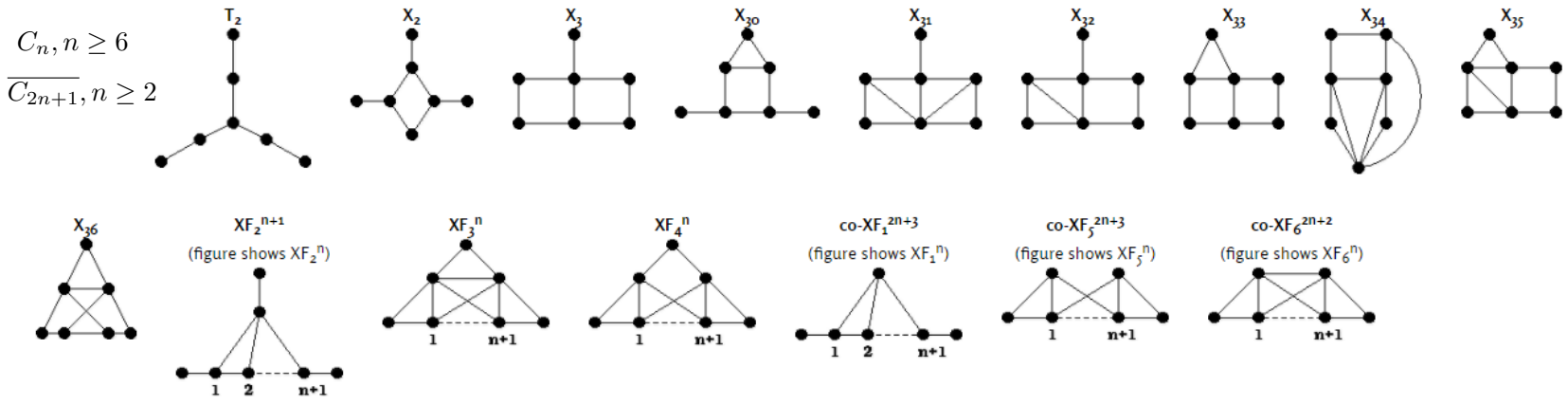


Figure source: http://graphclasses.org

The chromatic number of a cocomparability graph can be computed efficiently using Dilworth's theorem.

It follows that the lower bound

$$\sum_r \chi(G_{M,r})$$

can be computed in polynomial time.

Let us recall: Dilworth's theorem (1950)

Consider an arbitrary DAG (directed acyclic graph) $D = (V, A)$.

A **chain** in D is a set $\{v_1, \ldots, v_t\}$ of vertices such that there is a path from v_i to v_{i+1} in D, for all $i = 1, \ldots, t-1$.

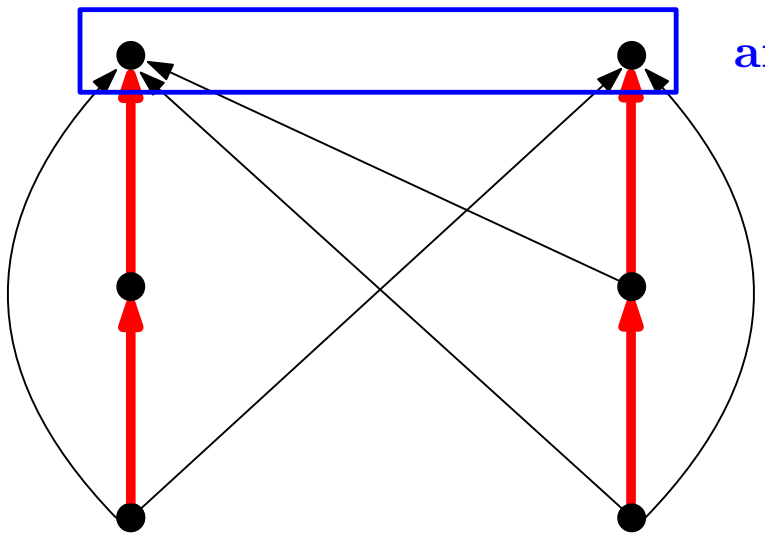
A **chain partition** of D is a family of vertex-disjoint chains $P = \{C_1, \ldots, C_p\}$ covering V.

An **antichain** in D is a set of vertices that are unreachable from each other.
(Equivalently, an independent set in the transitive closure of D.)

The **width** of D is the maximum size of an antichain.

Theorem (Dilworth, 1950)

In every DAG, the minimum size of a chain partition equals the width of D.



antichain of size 2

chain partition of size 2

Remark:
Dilworth's theorem is equivalent to the statement that every cocomparability graph is perfect.

- Minimum chain partitions of the DAG correspond to optimal colorings of the complement of the underlying graph of its transitive closure (which is a cocomparability graph).

An optimal chain partition can be computed in time $O(|V|^{5/2})$ by a solving a matching problem in a derived bipartite graph.

[Fulkerson 1956, Hopcroft-Karp 1973].

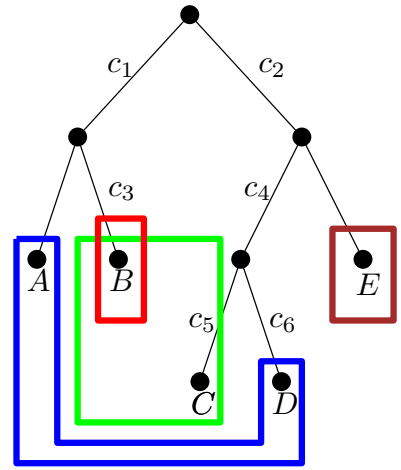
Back to the MCRS problem

M

	c_1	c_2	c_3	c_4	c_5	c_6
r_1	1	1		1		1
r_2	1		1			
r_3	1	1	1	1	1	
r_4		1				

M'

	c_1	c_2	c_3	c_4	c_5	c_6
R_1	1					
		1		1		1
R_2	1		1			
R_3	1		1			
		1		1	1	
R_4		1				



Hajirasouliha and Raphael claimed that the MCRS problem is **NP-hard**.

The proof was based on a reduction from the chromatic number problem in graphs and relied on the (erroneous) claim that every graph is a row-conflict graph.

They also presented an exponential-time algorithm based on graph coloring that, if correct, would imply that the **lower bound is always attained with equality**.

This is not the case.

We show:

Given a binary matrix M, it is NP-hard to determine whether $OPT_{MCRS}(M) = \sum_r \chi(G_{M,r})$.

Corollary: The MCRS problem is NP-hard.

We also showed that the MDCRS problem is NP-hard.

Reductions are from **3-edge-colorability in cubic graphs**.

A heuristic for MCRS

The lower bound $\mathrm{OPT}_{\mathrm{MCRS}}(M) \geq \sum_r \chi(G_{M,r})$ follows from the fact that

in every conflict-free row split of the input matrix M,

the rows replacing row r
can be used to produce a valid vertex coloring of $G_{M,r}$.

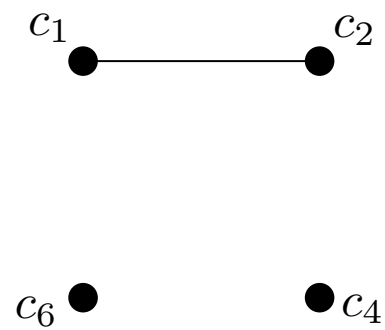
The difficulty in reversing this argument is due to the fact that

we cannot independently combine the splits of rows *r* of *M* according to optimal colorings of their conflict graphs, as new conflicts may arise.

M

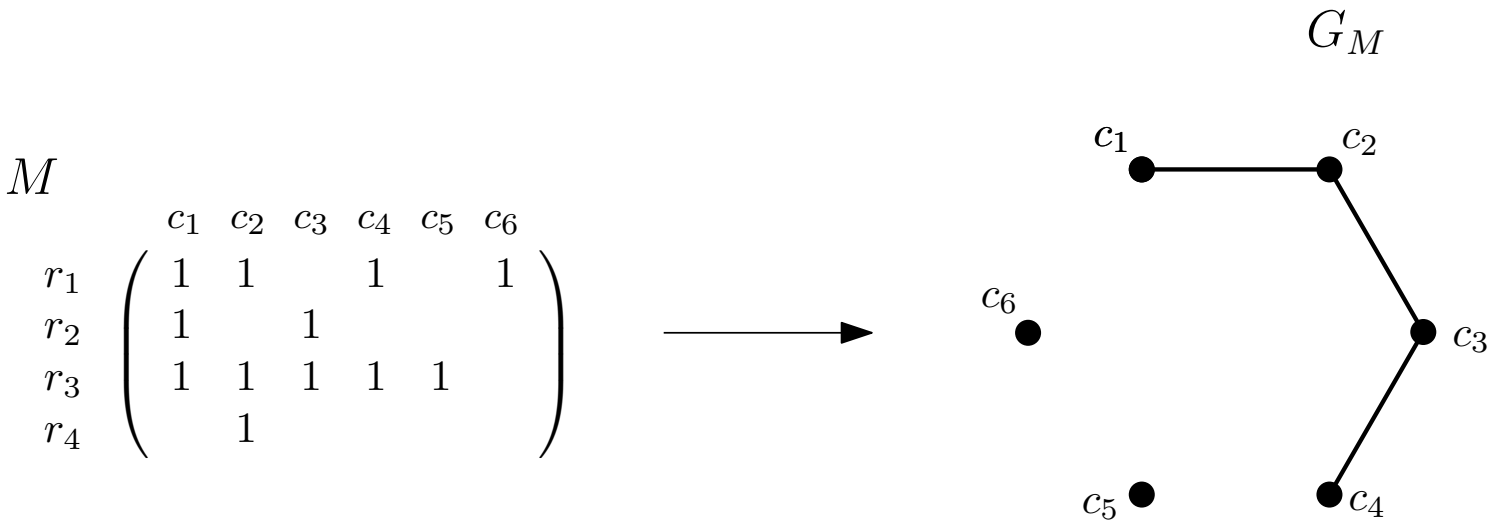
$$\begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{array} \begin{array}{c} \begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array} \\ \begin{pmatrix} 1 & 1 & & 1 & & 1 \\ 1 & & 1 & & & \\ 1 & 1 & 1 & 1 & 1 & \\ & 1 & & & & \end{pmatrix} \end{array}$$

G_{M,r_1}



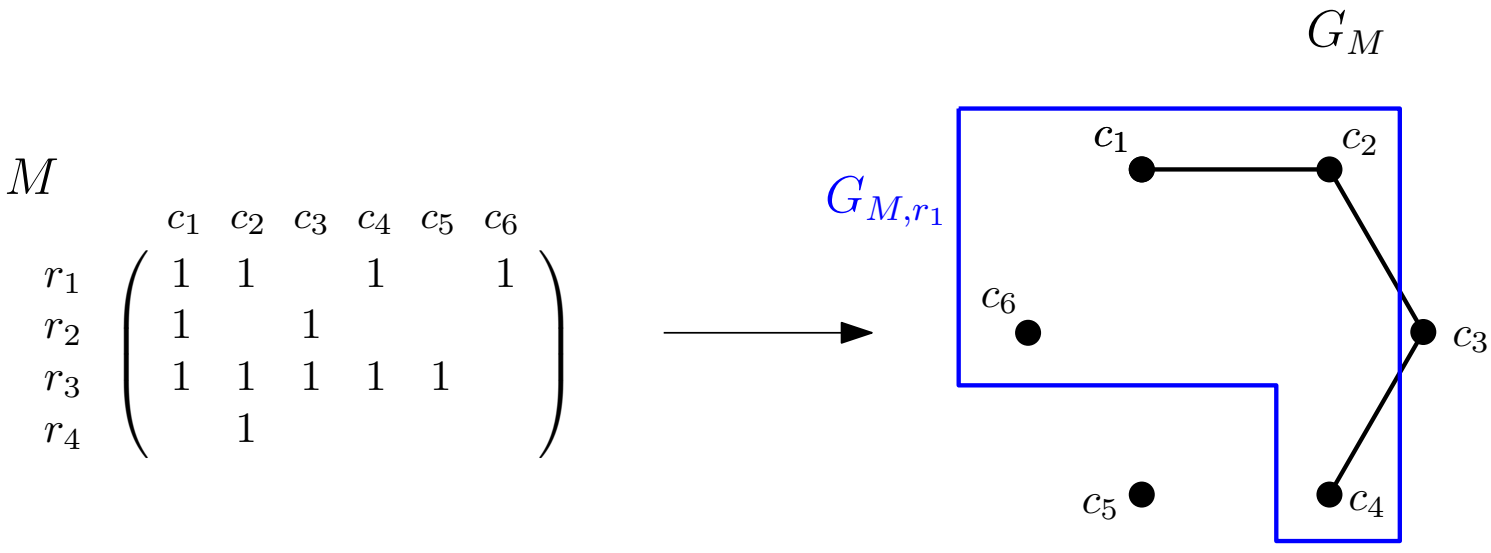
Given a binary matrix M, let us denote by G_M its **conflict graph**:

- we associate a vertex to each column of M
- edges represent conflicting column pairs.



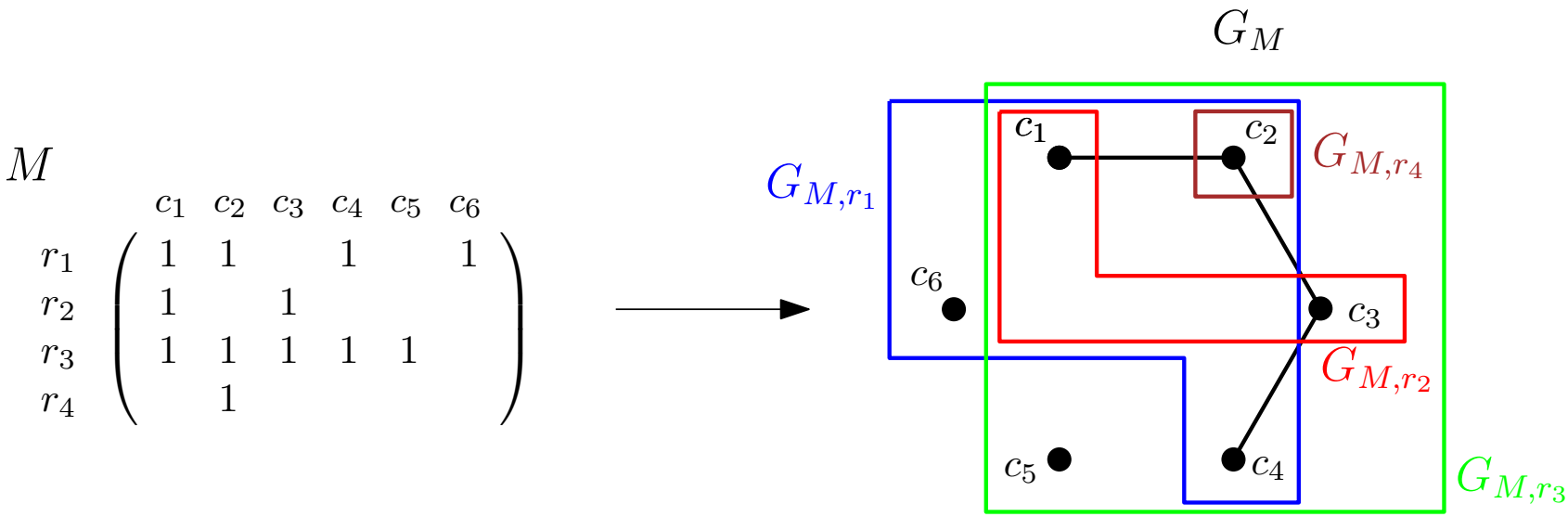
Given a binary matrix M, let us denote by G_M its **conflict graph**:

- we associate a vertex to each column of M
- edges represent conflicting column pairs.



Given a binary matrix M, let us denote by G_M its **conflict graph**:

- we associate a vertex to each column of M
- edges represent conflicting column pairs.



Can we color the graph G_M instead?

Difficulty:

Every graph $G = (V, E)$ of minimum degree at least 2 has

$$G \cong G_M \,,$$

where $M \in \{0, 1\}^{E \times V}$ is the edge-vertex incidence matrix of G.

It follows that optimally coloring conflict graphs is NP-hard.

Observation:
If two columns are not in conflict, then their support sets are either **disjoint** or **comparable (contained in one another)**.

Thus, $\overline{G_M} = H_d \cup H_c$,

"disjointness graph" $\cup$ "containment graph".

- If we keep only the "containment" part of it, then H_c is a comparability graph

  and an optimal coloring of $\overline{H_c}$ (a supergraph of G_M) can be found efficiently (using Dilworth's theorem).

- The subgraphs of $\overline{H_c}$ induced by $V(G_{M,r_i})$ are exactly the G_{M,r_i} graphs (hence the coloring of $\overline{H_c}$ gives rise to a valid row split).

So we have a **heuristic algorithm for the MCRS problem based on coloring cocomparability graphs**.

Part 2: results from ...

Hujdurović-Husić-M-Rizzi-Tomescu (WG 2017, arXiv)

Ademir Hujdurović, University of Primorska

Alexandru I. Tomescu, University of Helsinki

Romeo Rizzi, University of Verona

Edin Husić, ENS Lyon and University of Primorska (soon at LSE)

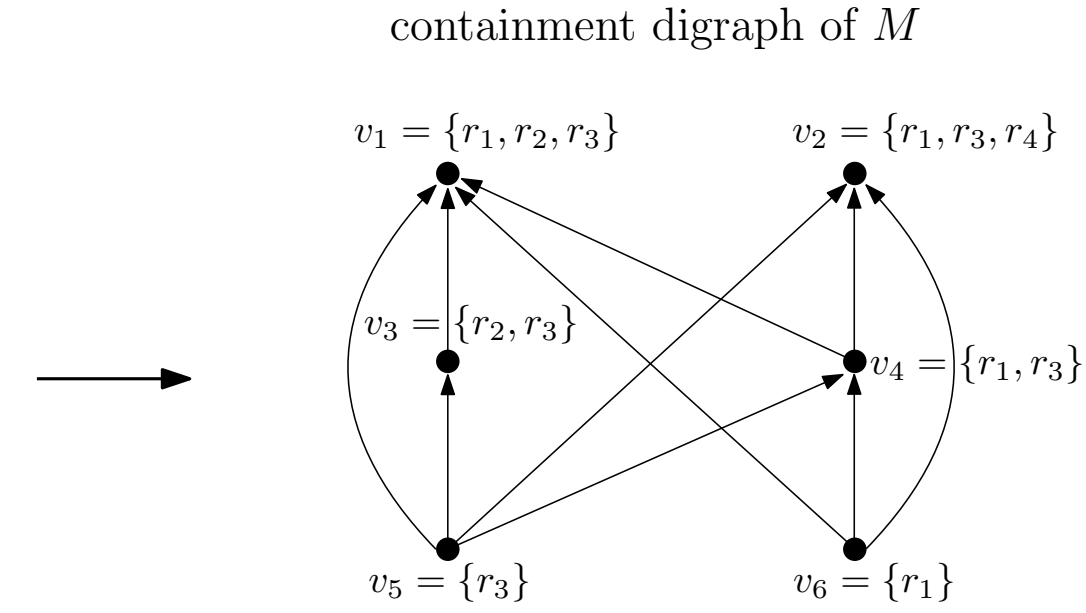
Formulations in terms of branchings in DAGs

A given binary matrix M is transformed into its **containment digraph** D_M:

- vertices = support sets of the columns
- arcs = containment relation

binary matrix M

$$\begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{array} \begin{array}{c} \begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array} \\ \begin{pmatrix} 1 & 1 & & 1 & & 1 \\ 1 & & 1 & & & \\ 1 & 1 & 1 & 1 & 1 & \\ & 1 & & & & \end{pmatrix} \end{array}$$

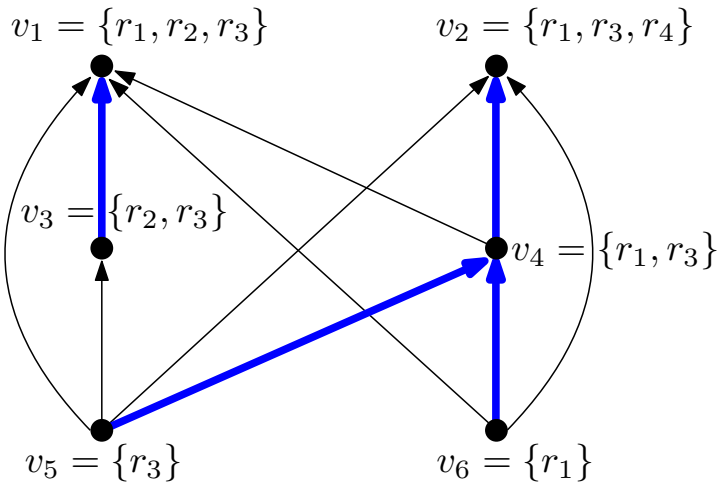


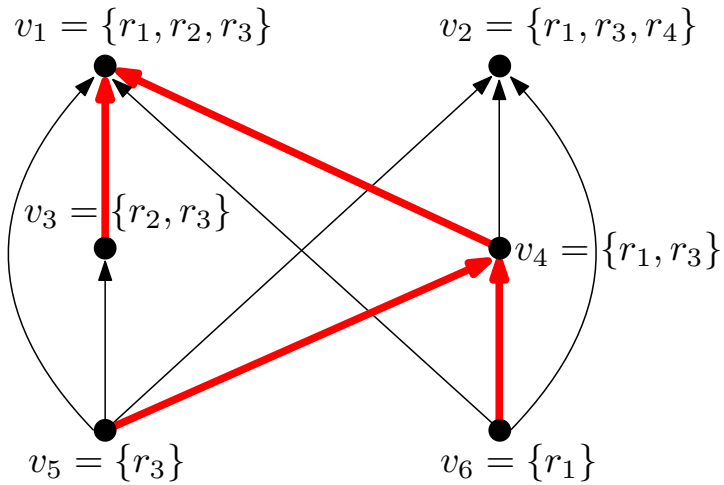
By construction, D_M is always a DAG (directed acyclic graph).

Definition

A **branching** in a DAG $D = (V, A)$ is a subset B of A such that in the digraph (V, B) for each vertex v there is at most one arc leaving v.

Two branchings of D_M:

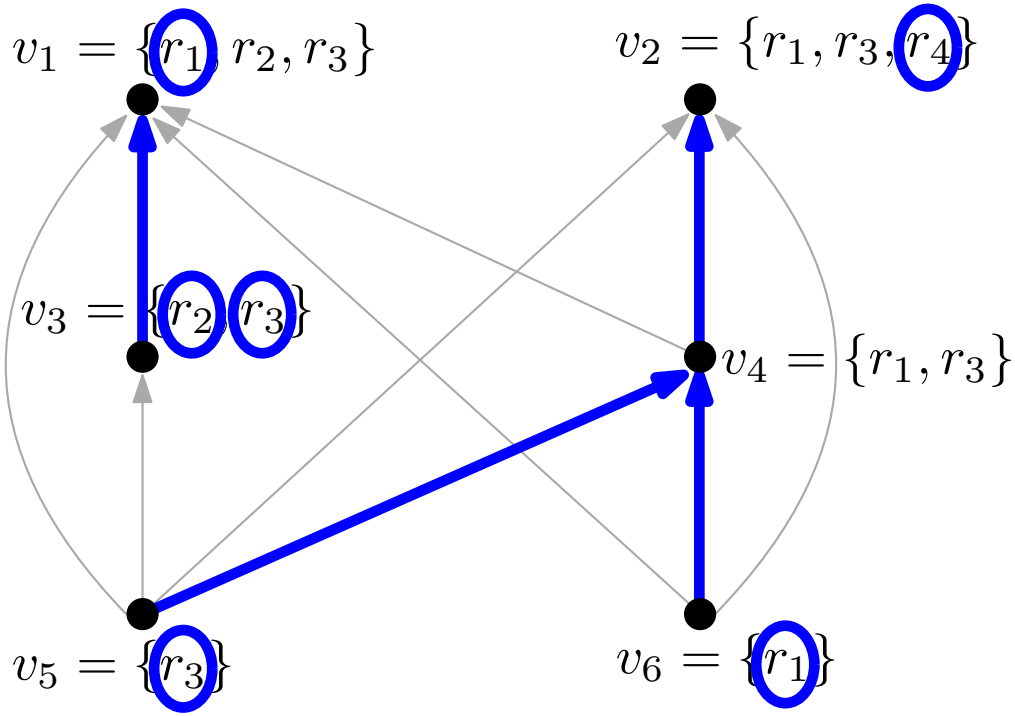




Given a branching B of D_M, we can count two quantities:

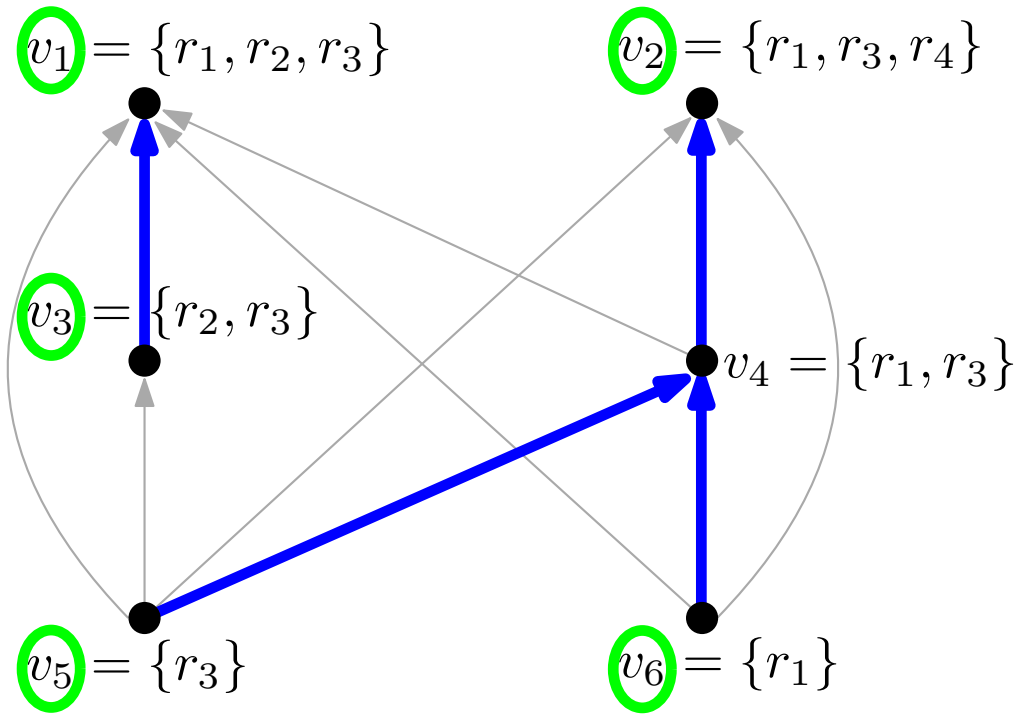
1. *The number of* **uncovered pairs**:

A pair (r, v) is **uncovered** if $r \in v \in V$ and $r \notin \cup\{w : w \in N_B^-(v)\}$



2. *The number of* **irreducible vertices***:*

A vertex $v \in V$ is **irreducible** if $v \neq \cup\{w : w \in N_B^-(v)\}$

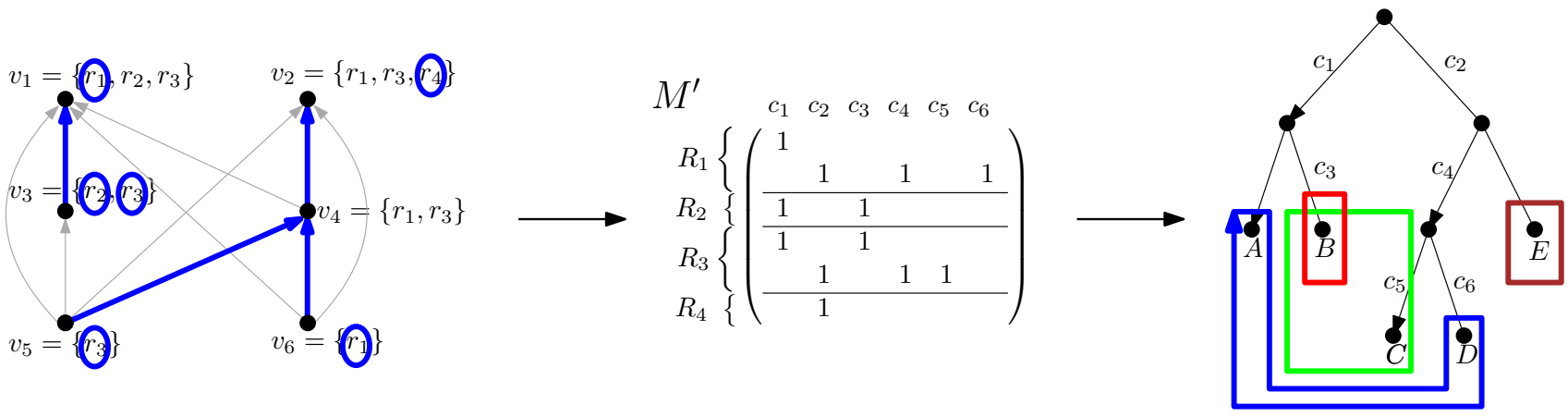


Theorem

The MCRS problem on a given binary matrix M

is equivalent to the problem of finding a branching in D_M

minimizing the **number of uncovered pairs**.

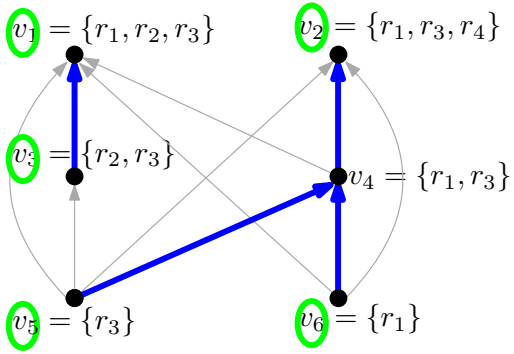


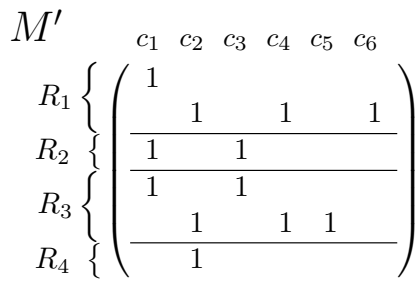
Theorem

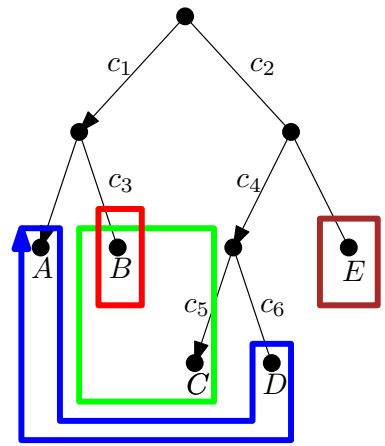
The MDCRS problem on a given binary matrix M

is equivalent to the problem of finding a branching in D_M

minimizing the **number of irreducible vertices**.





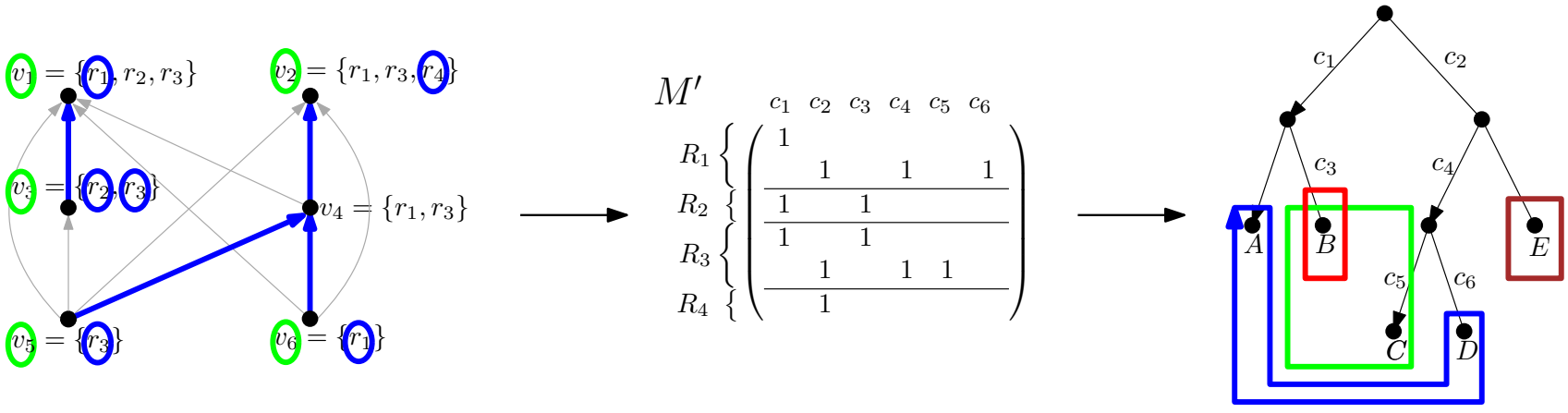


Theorem

The MDCRS problem on a given binary matrix M

is equivalent to the problem of finding a branching in D_M

minimizing the **number of irreducible vertices**.



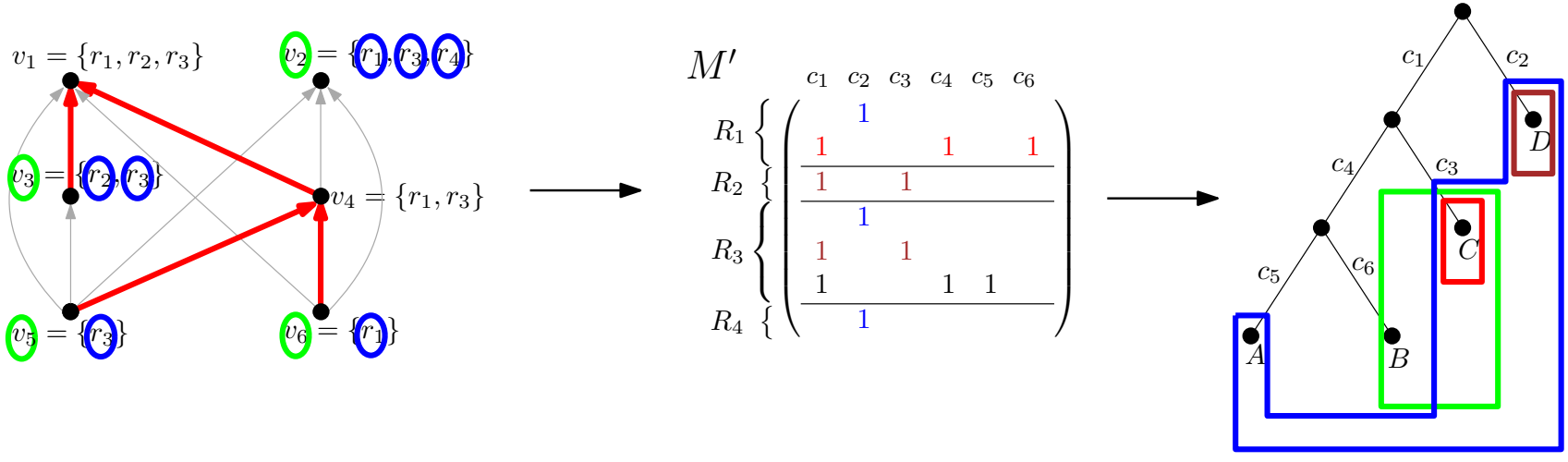
(6 uncovered pairs and 5 irreducible vertices)

Theorem

The MDCRS problem on a given binary matrix M

is equivalent to the problem of finding a branching in D_M

minimizing the **number of irreducible vertices**.



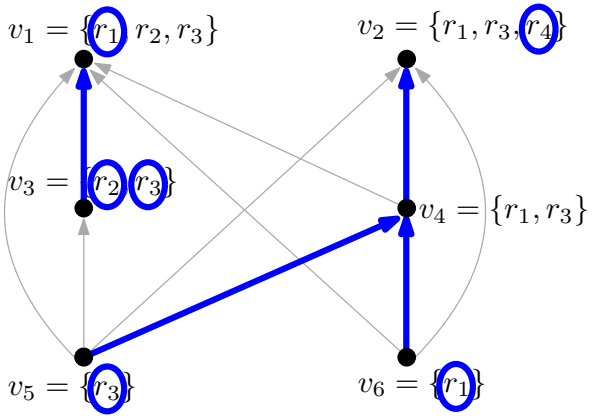
(7 uncovered pairs and 4 irreducible vertices)

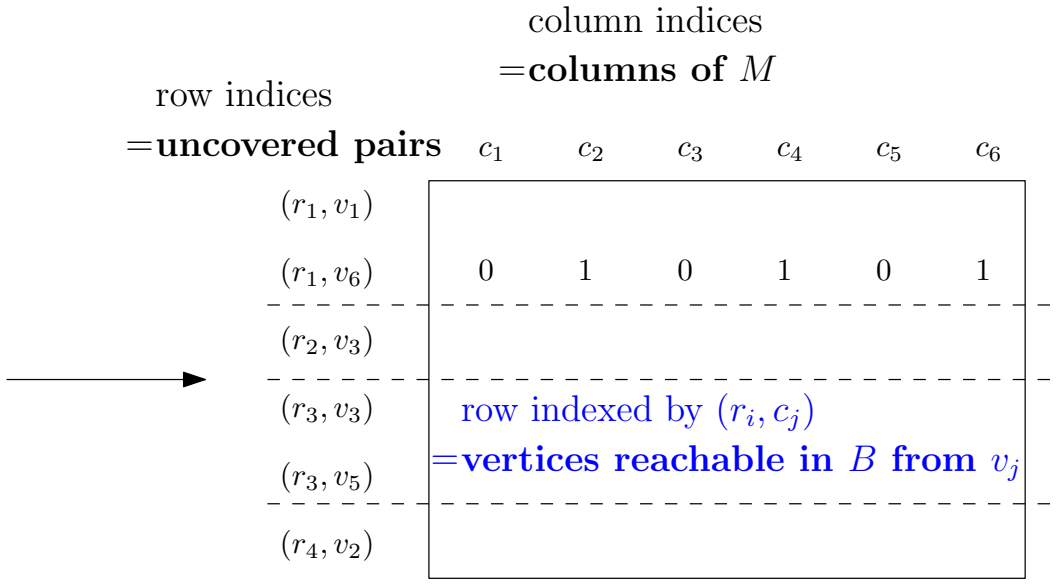
Theorem

The MCRS problem on a given binary matrix M

is equivalent to the problem of finding a branching in D_M

minimizing the **number of uncovered pairs**.





(In)approximability issues

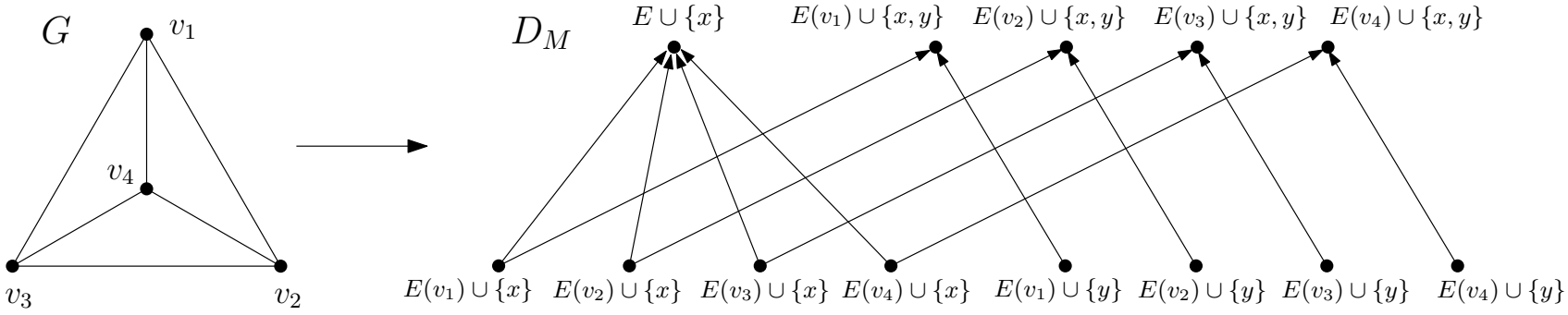
Theorem

The MCRS and the MDCRS problems are **APX-hard**, *even for instances of height* 2.

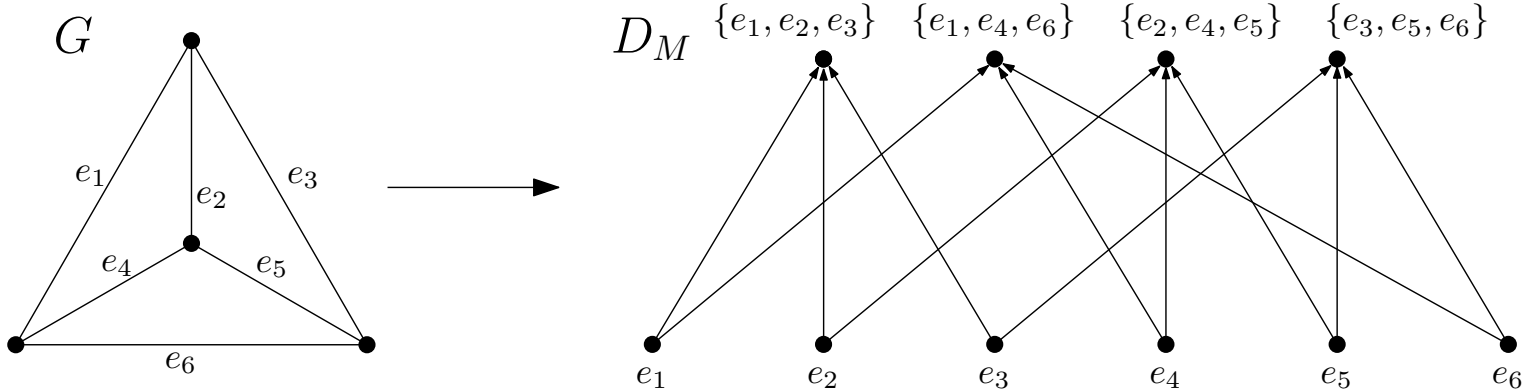
In other words, the two problems do not admit a PTAS (polynomial-time approximation scheme) unless P = NP .

Reductions are from **vertex cover in cubic graphs**,

For MCRS:



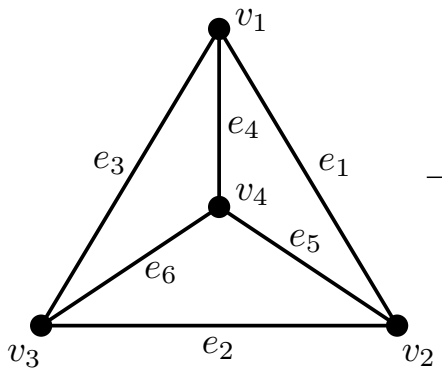
For MDCRS:



Initial transformations:

For MCRS:

$G = (V, E)$

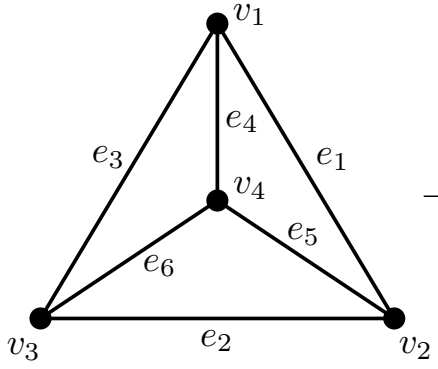


M

$$
\begin{array}{c}
 \\
v_1 \\ v_2 \\ v_3 \\ v_4 \\ r_1 \\ r_2 \\ r_3
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & c_1 & c_2 & c_3 \end{array} \\
\left(\begin{array}{ccccccc|ccc}
1 & & 1 & 1 & & & 1 & 1 & 1 \\
1 & 1 & & & 1 & & 1 & 1 & 1 \\
 & 1 & 1 & & & 1 & 1 & 1 & 1 \\
 & & & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
 & & & & & & 1 & & \\
 & & & & & & & 1 & \\
 & & & & & & & & 1
\end{array}\right)
\end{array}
$$

For MDCRS:

G

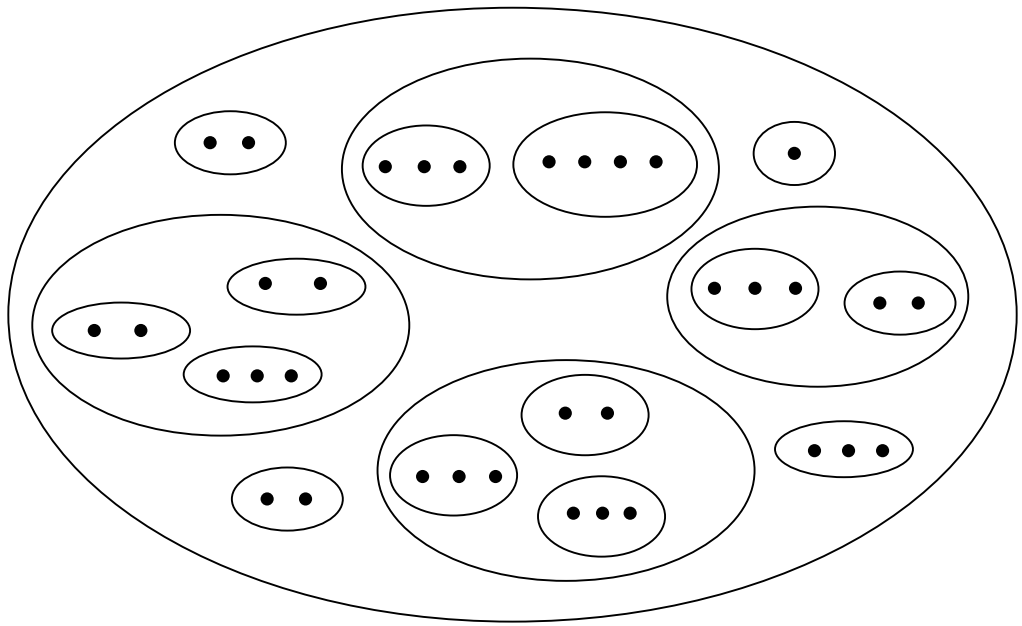


M

$$
\begin{array}{c}
 \\
v_1 \\ v_2 \\ v_3 \\ v_4 \\ r_1 \\ r_2 \\ r_3
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & c_1 & c_2 & c_3 & d_1 & d_2 & d_3 \end{array} \\
\left(\begin{array}{cccccc|ccc|ccc}
1 & & 1 & 1 & & & 1 & 1 & 1 & & & \\
1 & 1 & & & 1 & & 1 & 1 & 1 & & & \\
 & 1 & 1 & & & 1 & 1 & 1 & 1 & & & \\
 & & & 1 & 1 & 1 & 1 & 1 & 1 & & & \\
\hline
 & & & & & & 1 & & & 1 & & \\
 & & & & & & & 1 & & & 1 & \\
 & & & & & & & & 1 & & & 1
\end{array}\right)
\end{array}
$$

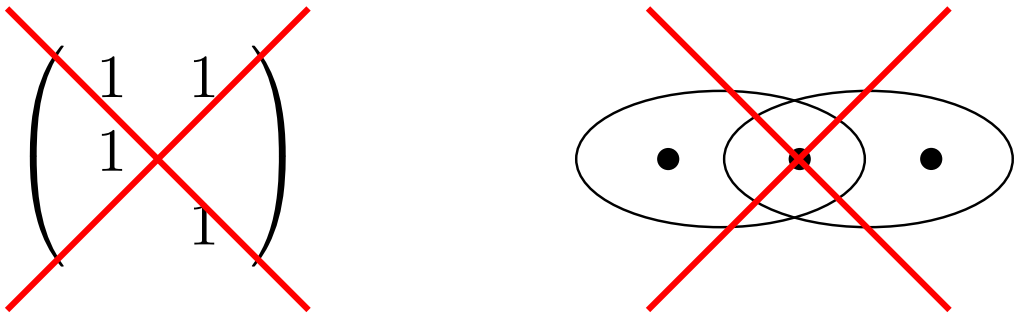
A 2-**approximation** to the MDCRS problem can be obtained using a connection with laminar families.

A hypergraph $\mathcal{H}$ is said to be **laminar** if every two hyperedges $e, f \in E(\mathcal{H})$ satisfy $e \cap f = \emptyset$, $e \subseteq f$, or $f \subseteq e$.



Facts:

1. A binary matrix M is conflict-free if and only if its **column hypergraph** $\mathcal{H}_M$ is laminar.
   - **1.1** vertices = rows
   - **1.2** hyperedges = support sets of the columns



2. (Schrijver, 2003)
   Every laminar hypergraph $\mathcal{H}$ satisfies $|E(\mathcal{H})| \leq 2|V(\mathcal{H})|$.

$\Rightarrow$ the trivial row splitting operation 2**-approximates MDCRS**

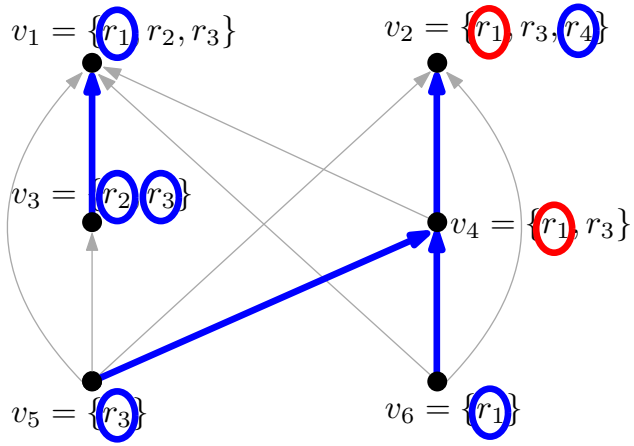
Constant-factor approximation algorithm for MCRS? Open.

Theorem

There is an h-approximation algorithm for the MCRS problem, where h is the height of D_M.

The algorithm: Return any branching.

- Fix an optimal branching B. Every pair (r, v) can be reached from some uncovered pair (r, v') of B.



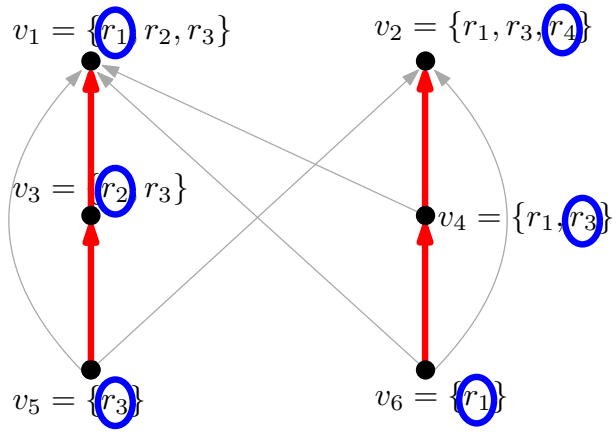
At most h pairs can be reached from any (r, v').

Theorem

There is a w-approximation algorithm for the MCRS problem, where w is the width of D_M.

The algorithm:

Return any branching arising from an optimal chain partition of D_M.

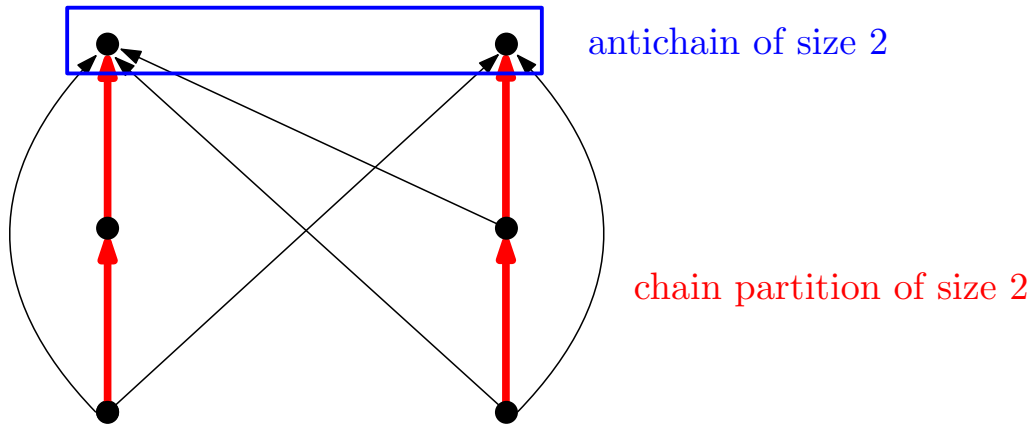


[Use Dilworth's theorem.]

- $\forall r$, there are $\leq w$ uncovered pairs of the form (r, v) (at most one in each path)

  $\Rightarrow$ total # of uncovered pairs is $\leq w \cdot$ number of r_i's

A generalization of Dilworth's theorem



The heuristic to the MCRS problem based on optimal colorings of the corresponding cocomparability graph

returned the solution corresponding to **any minimum chain partition of D_M.**

We show that it is possible to compute in polynomial time a chain partition P of D_M
minimizing the number of uncovered pairs.

In addition, $\#\text{chains in } P = \text{width}(D_M)$.

This improves on the previous heuristic.

The result is a consequence of a more general min-max result, which is a weighted generalization of Dilworth's theorem.

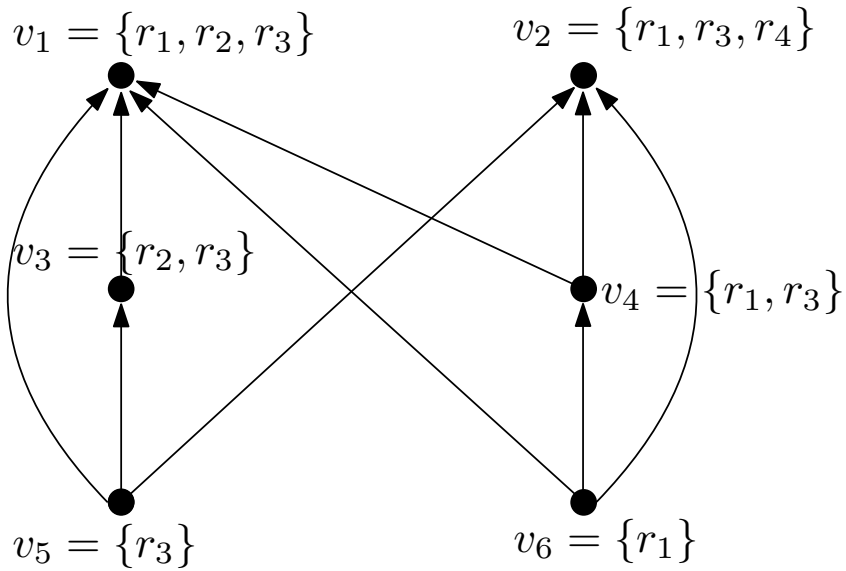
Given:

- a DAG $D = (V, A)$
- a **monotone weight function** $\pi : V \to \mathbb{Z}_+$
  (monotone: $\pi_u \leq \pi_v$ for every $(u, v) \in A$)

price of a chain C: $\Pi(C) = \max_{v \in C} \pi_v$.

price of a chain partition $P = \{C_1, \ldots, C_p\}$:

$$\Pi(P) = \sum_{i=1}^{p} \Pi(C_i) .$$



Given:

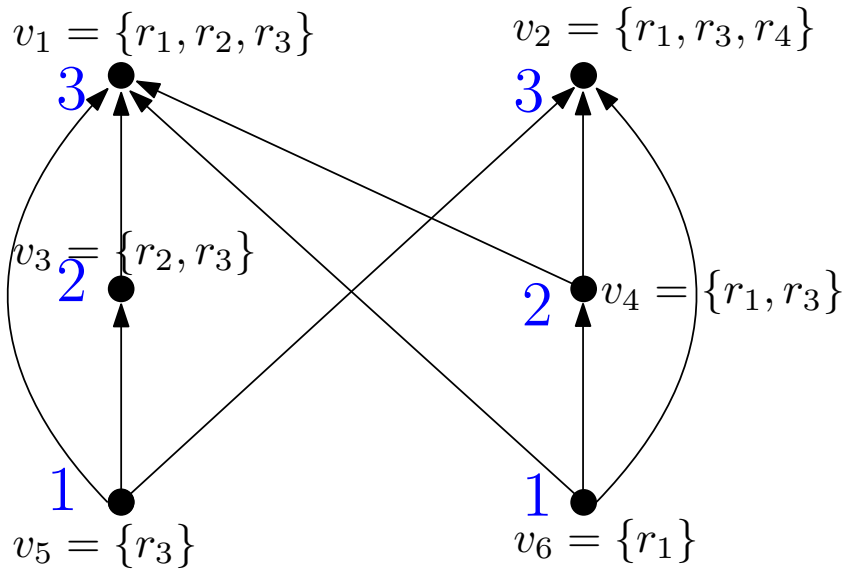
- a DAG $D = (V, A)$,
- a **monotone weight function** $\pi : V \to \mathbb{Z}_+$ (monotone: $\pi_u \leq \pi_v$ for every $(u, v) \in A$)

price of a chain C: $\Pi(C) = \max_{v \in C} \pi_v$.

price of a chain partition $P = \{C_1, \ldots, C_p\}$

$$\Pi(P) = \sum_{i=1}^{p} \Pi(C_i) .$$

$\pi_v = |v|$



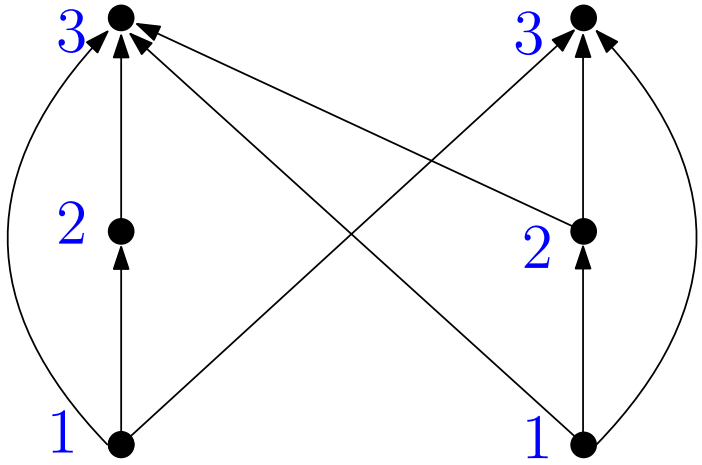
Given:

- a DAG $D = (V, A)$,
- a **monotone weight function** $\pi : V \to \mathbb{Z}_+$
  (monotone: $\pi_u \leq \pi_v$ for every $(u, v) \in A$)

price of a chain C: $\Pi(C) = \max_{v \in C} \pi_v$.

price of a chain partition $P = \{C_1, \ldots, C_p\}$:

$$\Pi(P) = \sum_{i=1}^{p} \Pi(C_i) .$$



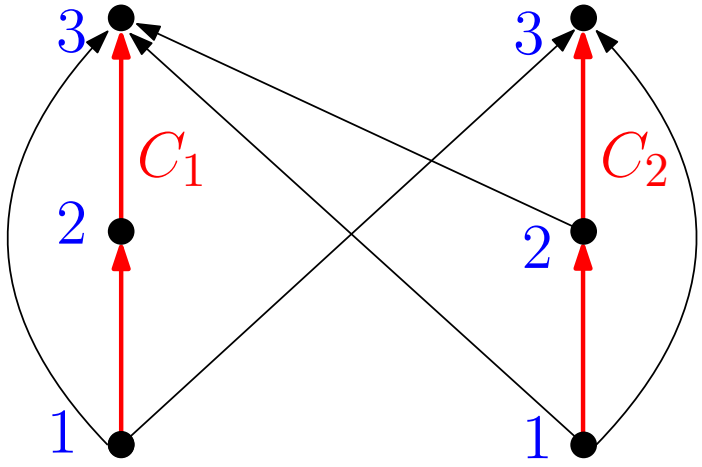
Given:

- a DAG $D = (V, A)$,
- a **monotone weight function** $\pi : V \to \mathbb{Z}_+$ (monotone: $\pi_u \leq \pi_v$ for every $(u, v) \in A$)

price of a chain C: $\Pi(C) = \max_{v \in C} \pi_v$.

price of a chain partition $P = \{C_1, \ldots, C_p\}$:

$$\Pi(P) = \sum_{i=1}^{p} \Pi(C_i) .$$



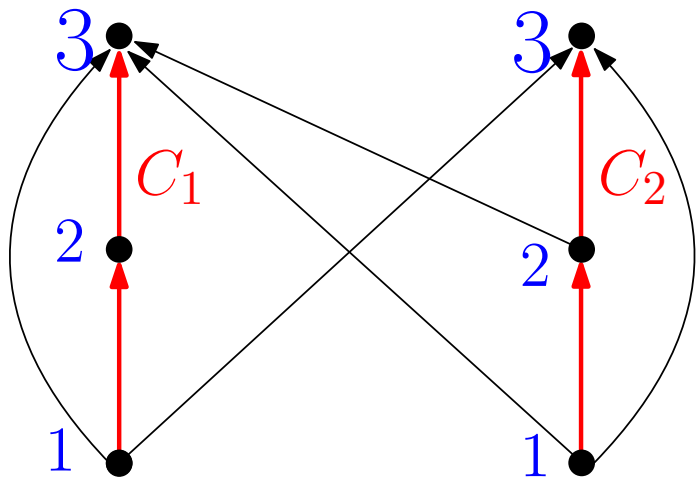
Given:

- a DAG $D = (V, A)$,
- a **monotone weight function** $\pi : V \to \mathbb{Z}_+$
  (monotone: $\pi_u \leq \pi_v$ for every $(u, v) \in A$)

price of a chain C: $\Pi(C) = \max_{v \in C} \pi_v$.

price of a chain partition $P = \{C_1, \ldots, C_p\}$:

$$\Pi(P) = \sum_{i=1}^{p} \Pi(C_i) .$$



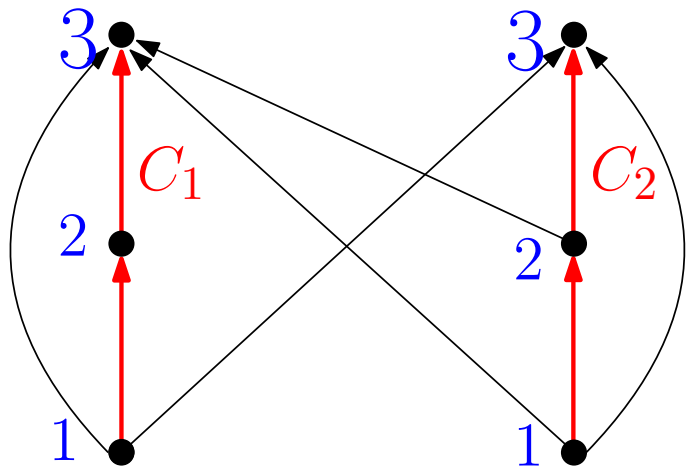
Given:

- a DAG $D = (V, A)$,
- a **monotone weight function** $\pi : V \to \mathbb{Z}_+$
  (monotone: $\pi_u \leq \pi_v$ for every $(u, v) \in A$)

price of a chain C: $\Pi(C) = \max_{v \in C} \pi_v$

price of a chain partition $P = \{C_1, \ldots, C_p\}$:

$$\Pi(P) = \sum_{i=1}^{p} \Pi(C_i) \,.$$

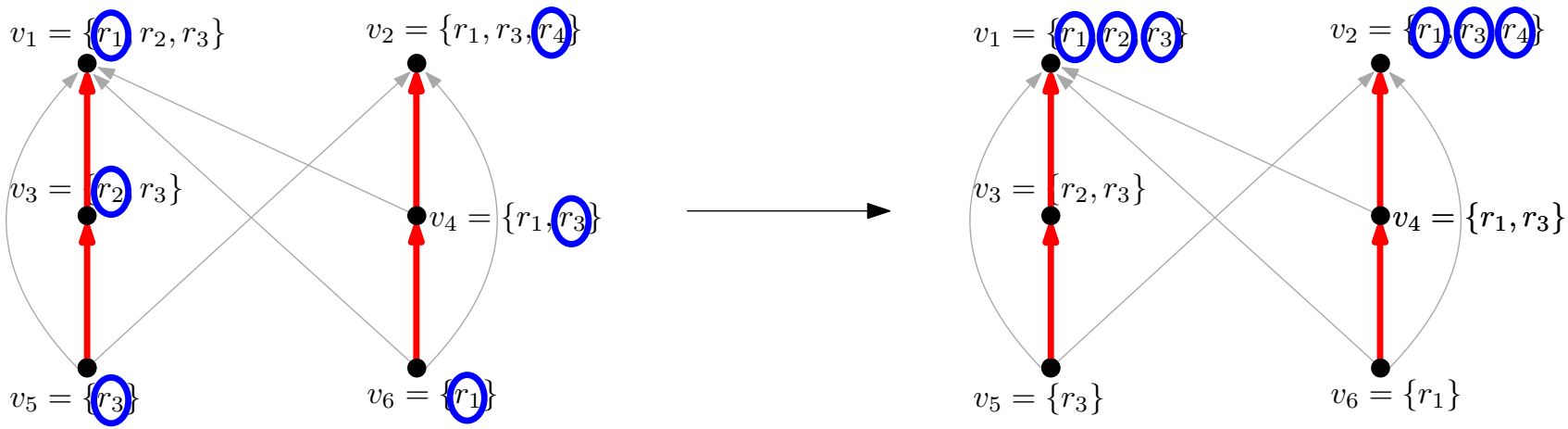


$\Pi(\{C_1, C_2\}) = 6$

Goal: given (D, π), find a chain partition of minimum price.

Why does this solve our problem?

Number of uncovered pairs = the price of the chain partition



A **tower of antichains** of D is a sequence

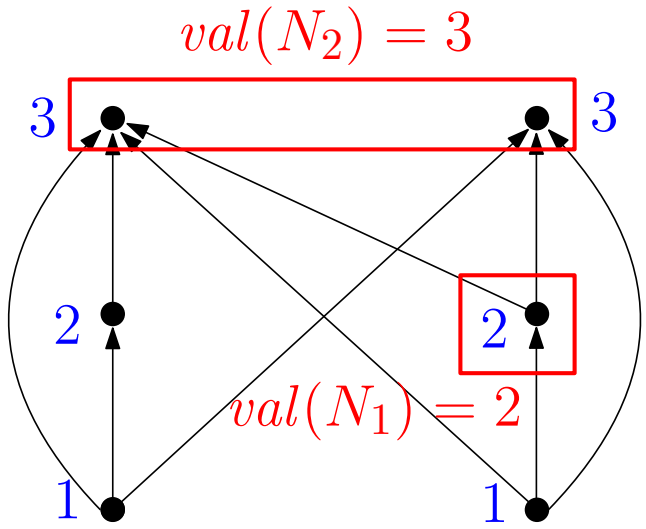
$$T = (N_1, N_2, \ldots, N_{width(D)})$$

of antichains with $|N_i| = i$.

value of an antichain N is $val(N) = \min_{v \in N} \pi_v$

value of a tower $T = (N_1, N_2, \ldots, N_{width(D)})$:

$$val(T) = \sum_{i=1}^{width(D)} val(N_i)\,.$$



$val(N_1, N_2) = 2 + 3 = 5$

Easy fact:

$\forall$ chain partition P and $\forall$ tower of antichains T:

$$\Pi(P) \geq val(T)\,.$$

Theorem

Every DAG D with a monotone weight function π has a chain partition

$$P = \{C_1, \ldots, C_{width(D)}\}$$

and a tower of antichains T such that $\Pi(P) = val(T)$. Moreover, such a pair (P, T) can be computed in time $O(|V(D)|^{7/2})$.

The proof is by induction.
The inductive step uses Dilworth's theorem.

The unit weight case?

In the case of **unit weights**, we have:

- $\Pi(P)$ = number of chains in P
- $val(T)$ = width of D

So we get Dilworth's theorem.

Remarks:

The minimum price partition problem becomes:

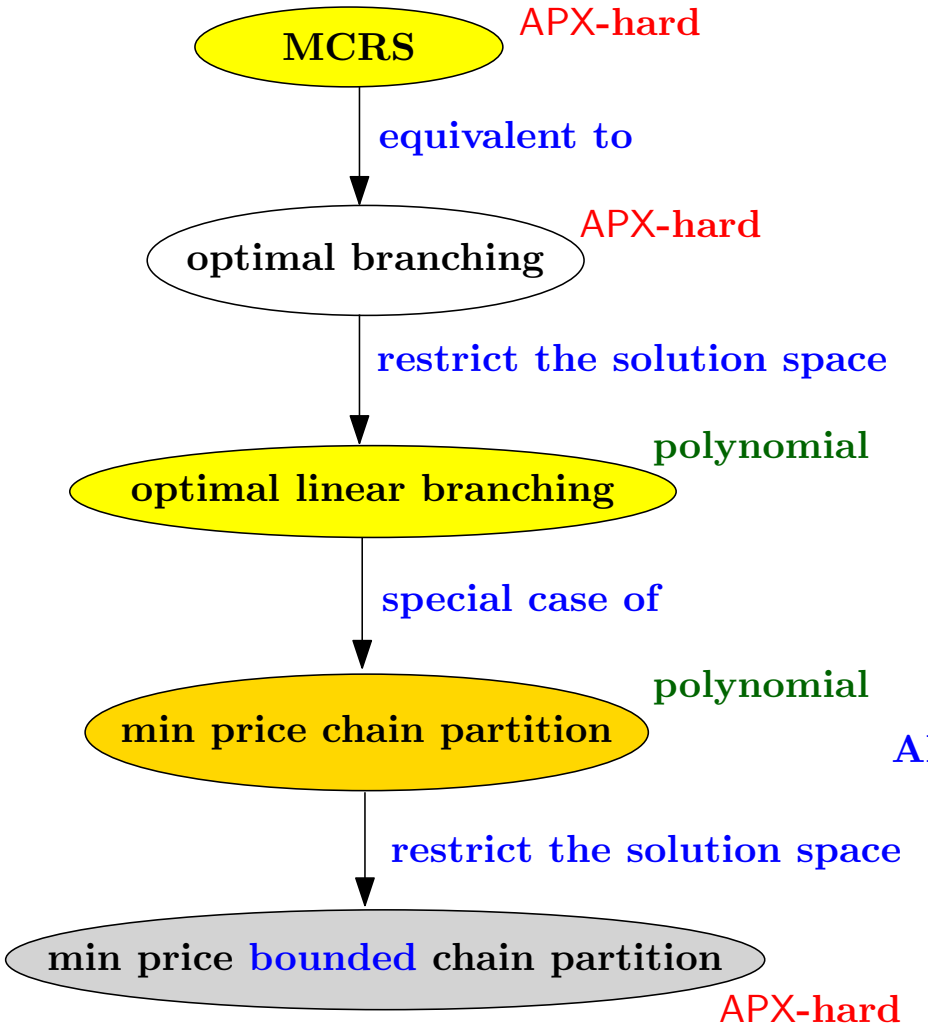
- **NP-hard** for non-monotone weight functions

  This follows from the fact that **Weighted Coloring** is NP-hard in interval graphs.

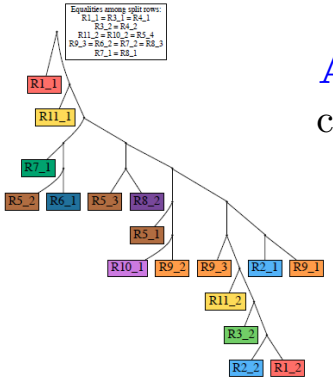
  [Escoffier, Monnot, and Paschos 2006]

- **APX-hard** if there is a bound on the length of the chains in the partition. [Moonen and Spieksma 2008]

Summary

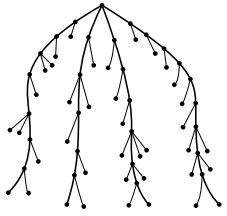


[Moonen-Spieksma, 2008]



APPLICATION:
cancer genomics

("**OR**" row splits)



"**XOR**" row splits

APPLICATION: ?

APPLICATION:
pallet-loading
(from a Dutch manufacturing company)

Open questions:

1. **(In)approximability** of MCRS?
   - How bad is the lower bound $\sum_r \chi(G_{M,r})$?

2. Complexity of MCRS and MDCRS for instances of **bounded width**?

3. (Suggested by Ekki Köhler:)
   Minimize $\max_r |\{(r, v) : (r, v) \text{ is uncovered}\}|$

4. Applications of the min price partition problem?

Thank you!

Merci!