Hybrid tractability of soft constraint problems

Standa Živný (Oxford)
(joint work with M. Cooper (Toulouse))

01/10/2010, Kiev, Ukraine

Outline

UNIVERSITY OF OXFORD

1. Introduction
2. Forbidden substructures
3. Polynomially-bounded search
4. Joint Winner Property (JWP)
5. Conclusion

Outline

1. **Introduction**
2. Forbidden substructures
3. Polynomially-bounded search
4. Joint Winner Property (JWP)
5. Conclusion

Problem

Given n variables $v_1, \ldots, v_n$ over domains $D_1, \ldots, D_n$,

$$\min_{v_1 \in D_1, \ldots, v_n \in D_n} \left(\sum_{i \in V} c_i(v_i) + \sum_{\{i,j\} \in E} c_{ij}(v_i, v_j) \right)$$

Problem

Given n variables $v_1, \ldots, v_n$ over domains $D_1, \ldots, D_n$,

$$\min_{v_1 \in D_1, \ldots, v_n \in D_n} \left(\sum_{i \in V} c_i(v_i) + \sum_{\{i,j\} \in E} c_{ij}(v_i, v_j) \right)$$

$$c_i : D_i \to \mathbb{R}_+ \cup \{\infty\}, \quad c_{ij} : D_i \times D_j \to \mathbb{R}_+ \cup \{\infty\}$$

Problem

Given n variables $v_1, \ldots, v_n$ over domains $D_1, \ldots, D_n$,

$$\min_{v_1 \in D_1, \ldots, v_n \in D_n} \Big(\sum_{i \in V} c_i(v_i) + \sum_{\{i,j\} \in E} c_{ij}(v_i, v_j) \Big)$$

$$c_i : D_i \to \mathbb{R}_+ \cup \{\infty\}, \quad c_{ij} : D_i \times D_j \to \mathbb{R}_+ \cup \{\infty\}$$

Binary VCSPs.

Problem

Given n variables $v_1, \ldots, v_n$ over domains $D_1, \ldots, D_n$,

$$\min_{v_1 \in D_1, \ldots, v_n \in D_n} \Big(\sum_{i \in V} c_i(v_i) + \sum_{\{i,j\} \in E} c_{ij}(v_i, v_j) \Big)$$

$$c_i : D_i \to \mathbb{R}_+ \cup \{\infty\}, \quad c_{ij} : D_i \times D_j \to \mathbb{R}_+ \cup \{\infty\}$$

Pairwise MRFs.

Problem

Given n variables $v_1, \ldots, v_n$ over domains $D_1, \ldots, D_n$,

$$\min_{v_1 \in D_1, \ldots, v_n \in D_n} \Big(\sum_{i \in V} c_i(v_i) + \sum_{\{i,j\} \in E} c_{ij}(v_i, v_j) \Big)$$

$$c_i : D_i \to \mathbb{R}_+ \cup \{\infty\}, \quad c_{ij} : D_i \times D_j \to \mathbb{R}_+ \cup \{\infty\}$$

Min-sum.

Problem

Given n variables $v_1, \ldots, v_n$ over domains $D_1, \ldots, D_n$,

$$\min_{v_1 \in D_1, \ldots, v_n \in D_n} \Big(\sum_{i \in V} c_i(v_i) + \sum_{\{i,j\} \in E} c_{ij}(v_i, v_j) \Big)$$

$$c_i : D_i \to \mathbb{R}_+ \cup \{\infty\}, \quad c_{ij} : D_i \times D_j \to \mathbb{R}_+ \cup \{\infty\}$$

NP-hard in general.

Terminology: CSP = CV

- variable = node
- domain value = label
- constraint/cost function = potential/quality function
- binary = pairwise
- Boolean = $\{0, 1\}$
- crisp/**hard** constraint: range $\{0, \infty\}$
- **soft** constraint: range $\mathbb{R}_+$
- CSP: all constraints are hard (or-and)
- VCSP: hard and soft constraints (min-sum)

Goal

Classes of VCSPs recognisable & solvable **exactly** in polynomial time.

Goal

Classes of VCSPs recognisable & solvable **exactly** in polynomial time.

Exact inference in (pair-wise) MRFs.

Tractability

Mostly 2 types of tractable classes:

- **language** - restrictions on all **cost functions**
  (ex: all cost functions are submodular)

- **structural** - restrictions on the **graph** $G = \langle V, E \rangle$
  (ex: G is a tree / has bounded tree-width)

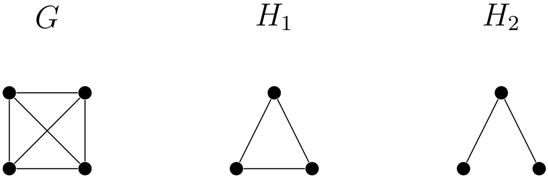
Hybrid tractable classes

Hybrid classes place restrictions on both cost functions and the structure (graph G).

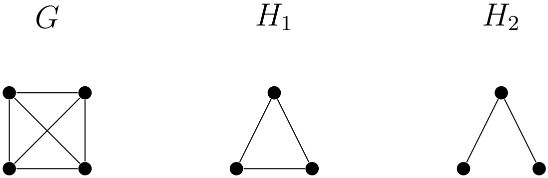
Outline

1. Introduction
2. **Forbidden substructures**
3. Polynomially-bounded search
4. Joint Winner Property (JWP)
5. Conclusion

Reminder: induced subgraphs

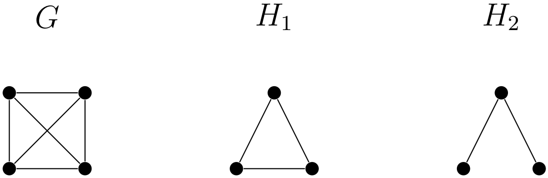


Reminder: induced subgraphs



Both H_1 and H_2 are subgraphs of G.
H_1 is an induced subgraph of G, but H_2 is not.

Reminder: induced subgraphs

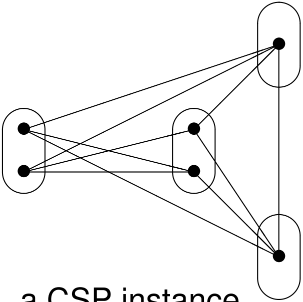


Both H_1 and H_2 are subgraphs of G.
H_1 is an induced subgraph of G, but H_2 is not.

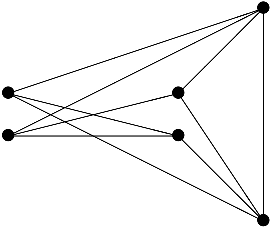
Given a class of graphs $\mathcal{F}$, a graph G is $\mathcal{F}$-free iff G does not contain any $F \in \mathcal{F}$ as an induced subgraph.

Simple example: CSPs (+soft unary)

Application of graph theory to the micro-structure.



a CSP instance
(edge=allowed)



its micro-structure

Using graph theory

Trivial observation:

- solving an n-var CSP instance = finding a clique of size n in the micro-structure
- equivalently, finding an independent set of size n in the micro-structure complement

Using graph theory

Corollary

CSPs with perfect, apple-free or fork-free micro-structure complement are tractable.

Using graph theory

Corollary

CSPs with perfect, apple-free or fork-free micro-structure complement are tractable.

Proof: Independent Set (IS) tractable for these classes.

Using graph theory

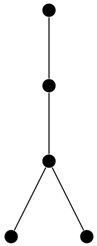
Corollary

CSPs with perfect, apple-free or fork-free micro-structure complement are tractable.

Proof: Independent Set (IS) tractable for these classes.

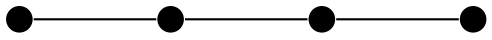
Also CSPs + soft unary: weighted IS.

Fork-free graphs



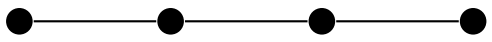
A fork

More on fork-free graphs

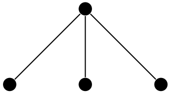


P_4

More on fork-free graphs

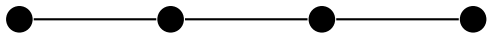


P_4

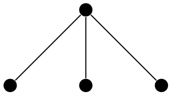


A claw

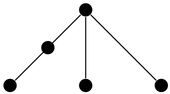
More on fork-free graphs



P_4

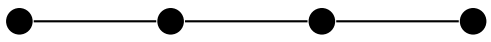


A claw

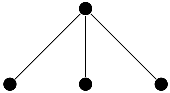


A fork

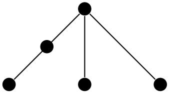
More on fork-free graphs



P_4



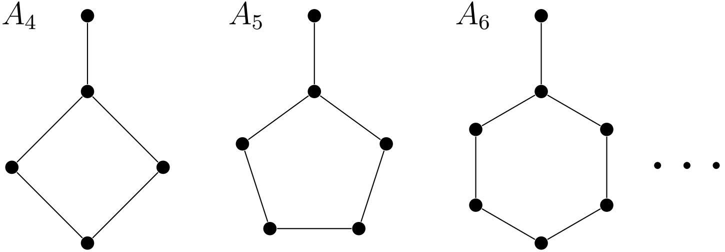
A claw



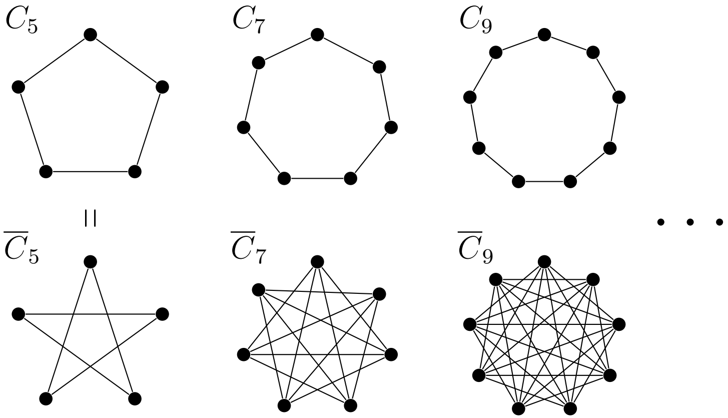
A fork

P_4-free $\cup$ claw-free $\subsetneq$ fork-free

Apple-free graphs



Perfect graphs



Summary of CSPs+soft unary

So far the objective function

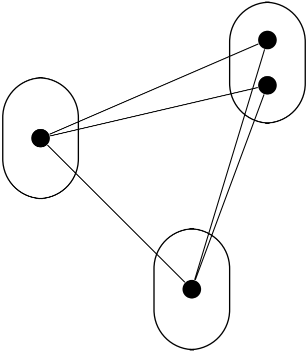
$$F = \sum_{i \in V} c_i(v_i) + \sum_{\{i,j\} \in E} c_{ij}(v_i, v_j)$$

can have only unary soft cost functions
(c_i can be soft; c_{ij} are are hard)

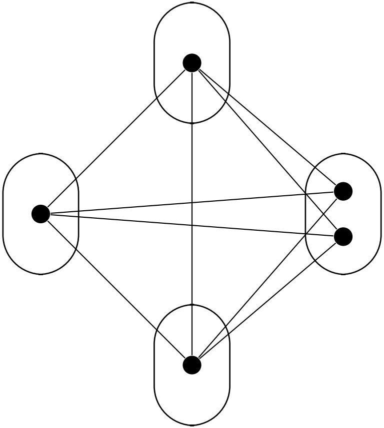
Outline

1. Introduction
2. Forbidden substructures
3. **Polynomially-bounded search**
4. Joint Winner Property (JWP)
5. Conclusion

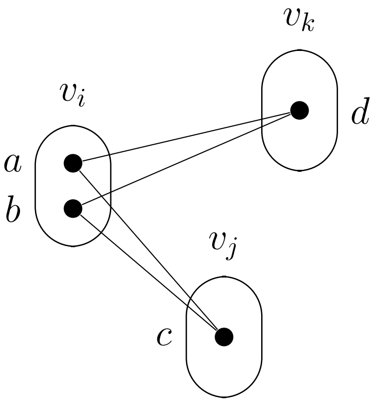
Polynomial number of solutions



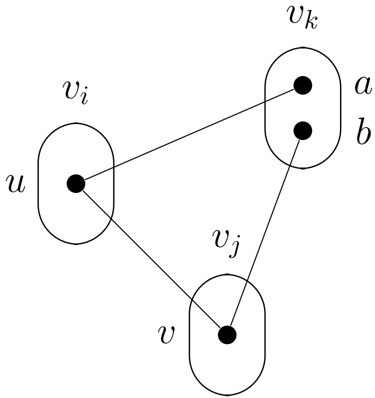
Polynomial number of solutions



Polynomial number of dead ends



Non-extendibility of BTP



Planar Max-Cut

Known example of a hybrid class: planar Max-Cut.

Structure not contractible to K_5, Boolean & symmetric cost functions. (Unary soft cost functions NP-hard.)

Hybrid tractability

- language classes: **algebra**
- structural classes: **graph theory**
- hybrid classes: we propose to study instances defined by **properties of subproblems** of size k

  First interesting (genuinely hybrid) case:
  binary constraints and $k = 3$

Outline

1. Introduction
2. Forbidden substructures
3. Polynomially-bounded search
4. **Joint Winner Property (JWP)**
5. Conclusion

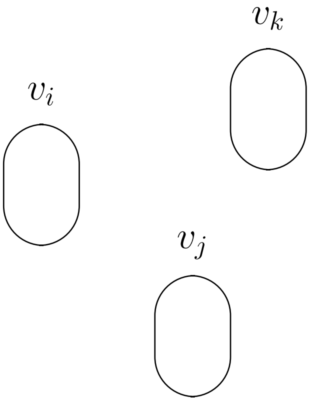
Joint-Winner Property (JWP)

v_k

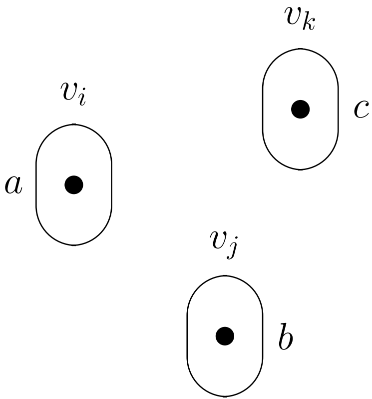
v_i

v_j

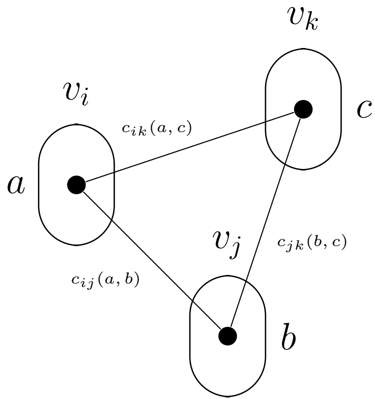
Joint-Winner Property (JWP)



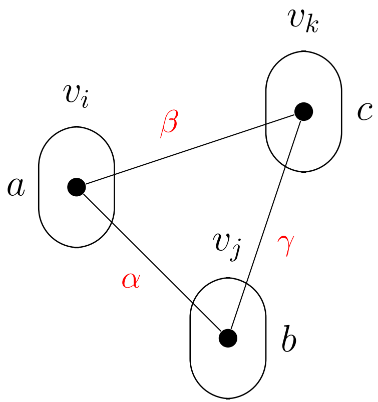
Joint-Winner Property (JWP)



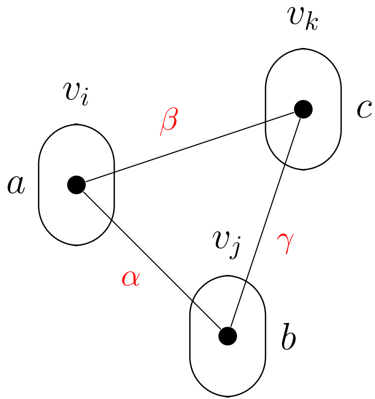
Joint-Winner Property (JWP)



Joint-Winner Property (JWP)

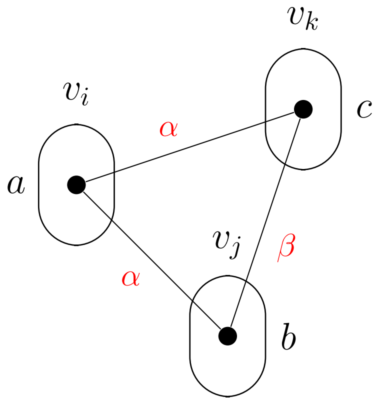


Joint-Winner Property (JWP)



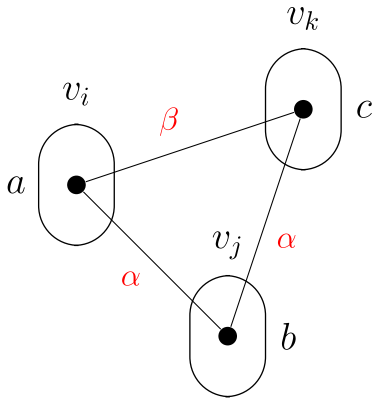
$$(\alpha \geq \min(\beta, \gamma)) \ \wedge \ (\beta \geq \min(\alpha, \gamma)) \ \wedge \ (\gamma \geq \min(\alpha, \beta))$$

Joint-Winner Property (JWP)



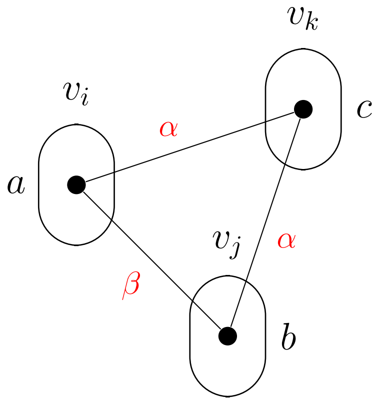
$$\alpha \leq \beta$$

Joint-Winner Property (JWP)



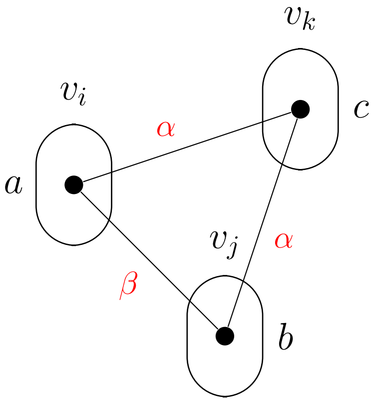
$$\alpha \leq \beta$$

Joint-Winner Property (JWP)



$$\alpha \leq \beta$$

Joint-Winner Property (JWP)



$\alpha \leq \beta$

JWP is conservative, i.e. includes all unary cost functions.

Example 1: ALLDIFF + soft unary

Recruiting a different person for each of n jobs (where for each job there is an ordered short-list of candidates).

Example 2: SOFTALLDIFF + soft unary

Choosing someone to be in charge of each of n courses (where for each course there is an ordered short-list of candidates). We would prefer to avoid (although this is not disallowed) to put the same person in charge of more than one course.

Example 3: Machine schedulling

$t_i(m) =$ execution time of job i on machine m.

Let S_m be the set of jobs assigned to machine m. On the same machine we sort the jobs S_m in increasing order of $t_i(m)$. The total waiting time of all jobs

$$T = \sum_{m=1}^{d} \Big(\sum_{i \in S_m} t_i(m) + \sum_{\substack{i,j \in S_m \\ i<j}} \min(t_i(m), t_j(m)) \Big)$$

This is a binary VCSP with unary constraints $c_i(m) = t_i(m)$ (the execution time of the jobs) and binary constraints

$$c_{ij}(m, m') = \begin{cases} \min(t_i(m), t_j(m)) & \text{si } m = m' \\ 0 & \text{otherwise} \end{cases}$$

This VCSP satisfies the JWP because
$\min(t_i(m), t_j(m)) \geq \min(\min(t_i(m), t_k(m)), \min(t_j(m), t_k(m)))$.

Example 4: MAX-2SAT

Given 2 -CNF formula ϕ without repeated clauses:

$$(l \vee l_1) \in \phi \quad \wedge \quad (l \vee l_2) \in \phi \quad \Rightarrow \quad (l_1 \vee l_2) \in \phi,$$

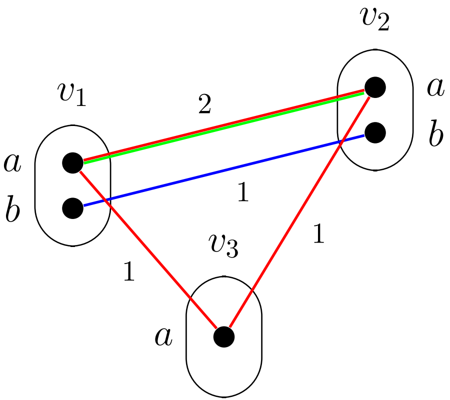
where l, l_1, l_2 are three literals of distinct variables.

Consequences of the JWP

Lemma

1. the set of micro-structure edges $\{(i,a),(j,b)\}$ s.t. $c_{ij}(a,b) \geq \beta$ decompose into non-intersecting cliques
2. for each pair of cliques C_α, C_β:
   if $C_\alpha \cap C_\beta \neq \emptyset$ and $\alpha \leq \beta$, then $C_\beta \subseteq C_\alpha$

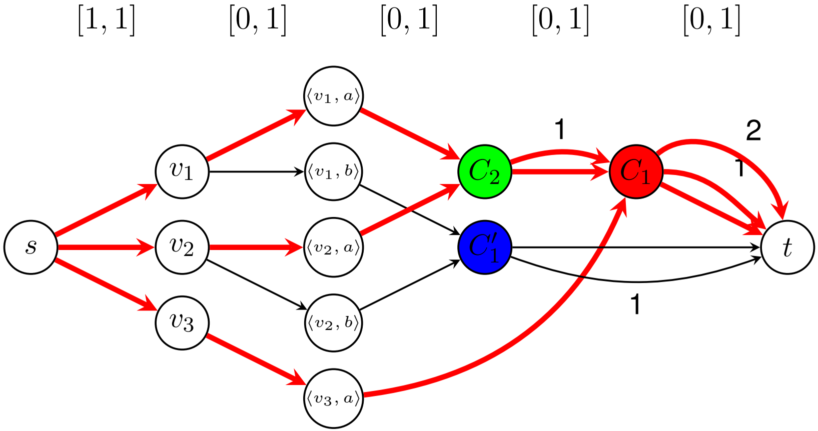
Example



(no edge=cost 0)

Algorithm

Optimal solution = minimum-cost maximum-flow



Number of cliques

An obvious upper bound $O(n^2d^2)$ on the number of different costs and hence cliques.

Number of cliques

An obvious upper bound $O(n^2d^2)$ on the number of different costs and hence cliques.

Lemma

There are at most $2dn - 1$ maximal cliques.

Complexity

- recognition in $O(n^3 d^3)$
- network can be built in $O(n^3 d^3)$ via $O(nd)$-times DFS
- network has $nd + nd + n + 2 = O(nd)$ vertices
- using the successive shortest path algorithm to determine a minimum-cost maximum flow in a network, using Fibonacci heaps: $O(n^3 d^2)$

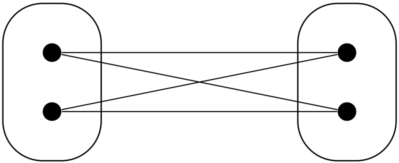
Z-property

JWP restricts three variable at once, and there are no restrictions on the unary cost funtions.

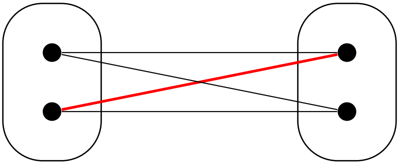
Any restrictions on pairs of variables?

Z-property

Z-property

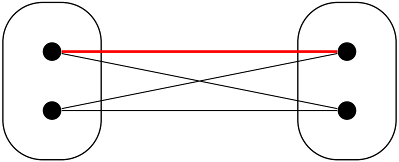


Z-property



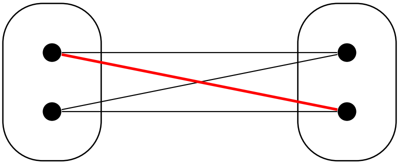
Cost of the red one < then the other three costs.

Z-property



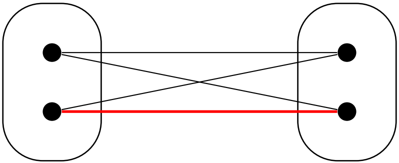
Cost of the red one < then the other three costs.

Z-property



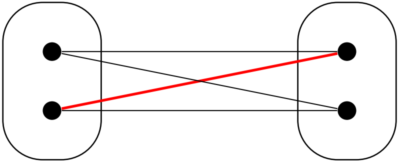
Cost of the red one < then the other three costs.

Z-property



Cost of the red one < then the other three costs.

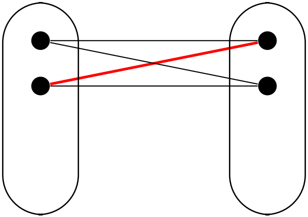
Z-property



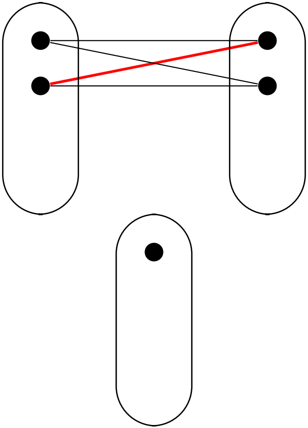
Cost of the red one < then the other three costs.

A **Z** breaks the clique decomposition $:-()$

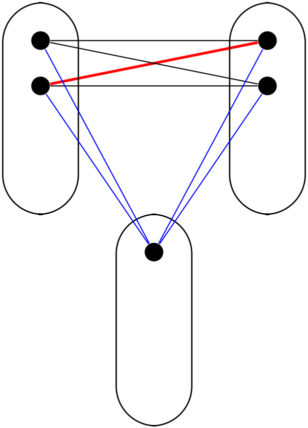
Z-freeness



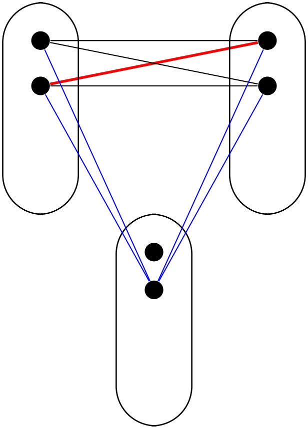
Z-freeness



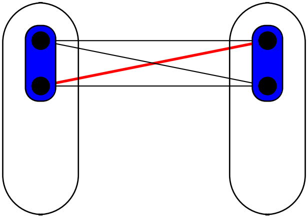
Z-freeness



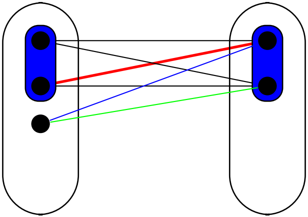
Z-freeness



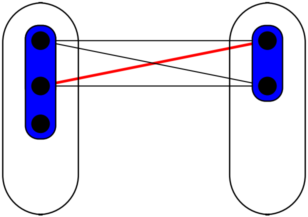
Z-freeness



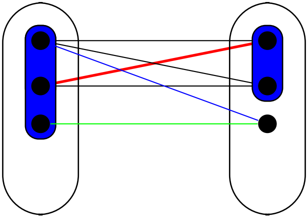
Z-freeness



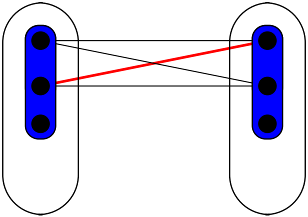
Z-freeness



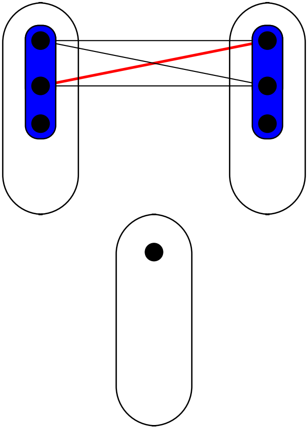
Z-freeness



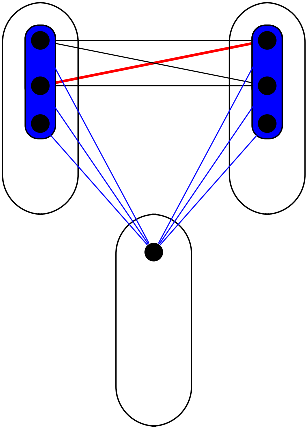
Z-freeness



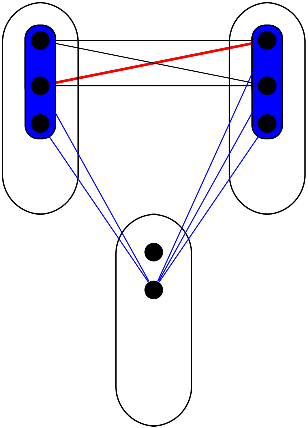
Z-freeness



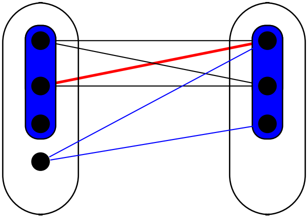
Z-freeness



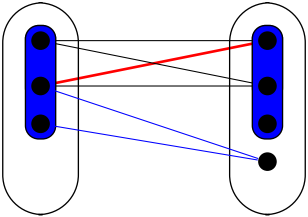
Z-freeness



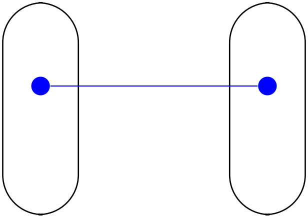
Z-freeness



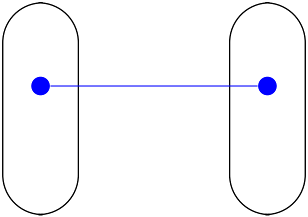
Z-freeness



Z-freeness



Z-freeness



New unary and binary costs depending on optimal assignments.

Analysis

- $O(n^2d^4)$ to detect all **Z**s
- addition of every new value is $O(d^2)$, can happen only $O(nd)$ times
- overall: $O(n^2d^4 + nd^3) = O(n^2d^4)$

Overall complexity

- Z-freeness: $O(n^2 d^4)$
- max-flow: $O(n^3 d^2)$
- **total**: $O(n^3 d^4)$

Theorem

JWP is recognisable in $O(n^3 d^3)$ and solvable in $O(n^3 d^4)$.

Maximality

UNIVERSITY OF OXFORD

Theorem

Let α, β, γ be three fixed costs not satisfying JWP. Then the following class of VCSPs in NP-hard: every triangle either satisfies JWP or has costs $\{\alpha, \beta, \gamma\}$.

Generalisation

The same min-cost max-flow algorithm to minimise

$$f(A) \;=\; f_1(|C_1 \cap A|) \;+\; \ldots \;+\; f_r(|C_r \cap A|)$$

$C_1, \ldots, C_r$ non-intersecting sets of (variable,value) assignments, $f_1, \ldots, f_r$ non-decreasing functions with non-decreasing derivatives and $|C_i \cap A|$= number of assignments in the solution A which belong to C_i.

Example

(Max-)CSP instances in which the "nogoods" (considered as sets of (variable,value) assignments) do not intersect.

Outline

Summary

- binary VCSPs (=pairwise MRFs)
- hybrid tractability
- classes defined via fobidden substructures
- joint winner property (JWP)

Problems

- Applications: fewer cliques, Z-free,...
- Better non-binary generalisation?
- Other properties of size-k subproblems?

Thank you

Full version: `arXiv:1008.4071`

`http://zivny.cz/`

Questions?