Parameterising the Complexity of Planning by the Number of Paths in the Domain-transition Graphs

Christer Bäckström

Linköping University

Outline

- Introduction
- Parameterised complexity
- Planning formalism and terminology
- Parameters used and treewidth
- Main results for plan decision
- Results for plan optimisation
- Application to delete relaxation heuristics
- Discussion

Introduction

Previous Results:

- Complexity results for various restrictions on the domain-transition graphs

- Complexity results for various restrictions on the causal graph

- Very few results on combined restrictions on both graph types

- Usually only binary: having a restriction or not

In This Paper:

- Exploits combination of properties of the causal graph and the domain-transition graphs

- Quantitative properties of the graphs, rather than binary ones, by exploiting parameterised complexity theory

Parameterised Complexity

Let n be instance size and k some parameter of the instance

Tractable in the standard way:

Time $O(n^c)$ for some constant c

Fixed-parameter tractable (fpt):

Time $O(f(k) \cdot n^c)$ for some function f and constant c

The expression is *separable* into

- a *hard* part $f(k)$ and
- an *easy* part n^c

Example:

$2^{k^2}n^3$ is fpt, but neither n^k nor $k^{\log n}$ are fpt.

FPT is the class of all fpt problems

There are harder classes and a completeness theory

$$\mathbf{P} \subseteq \mathbf{FPT} \subseteq \mathbf{W}[1] \subseteq \mathbf{W}[2] \subseteq \ldots$$

No hardness results in this paper.

Planning (SAS$^+$ formalism):

Planning instance $\mathbb{P} = \langle V, D, A, s_I, s_G \rangle$ where

- Variables $V = \{v_1, \ldots, v_n\}$, each w. finite domain $D(v_i)$
- Actions A, each with precondition pre(a) and effect eff(a)
- Initial state s_I and goal s_G

Let $V' \subseteq V$ and ω an action sequence

- $\mathbb{P}[V']$ is the projection of $\mathbb{P}$ to V'
- $\omega[V']$ is the subsequence of actions affecting variables in V'

Transition Graphs:

The *transition graph* for $\mathbb{P}$ is the labelled digraph
$TG = \langle S, E \rangle$ where

- S is the state space for $\mathbb{P}$
- $\langle s, a, t \rangle \in E$ if action a is from s to t

The *domain-transition graph (DTG)* for a variable v is
DTG(v) $= TG(\mathbb{P}[v])$

Causal Graphs:

The *causal graph* for $\mathbb{P}$ is the digraph
$CG(\mathbb{P}) = \langle V, E \rangle$ where

- V are the variables
- $\langle u, v \rangle \in E$ if there is some action a such that either
  - $u \in \text{eff}(a)$ and $v \in \text{pre}(a)$ or
  - $u \in \text{eff}(a)$ and $v \in \text{eff}(a)$

Parameters and Assumptions

Three instance parameters will be considered:

d : Domain size of the variables

k : Max. number of paths in each DTG

w : Treewidth of the causal graph

These are all fpt to check

All results assume acyclic DTGs.
However, rather a consequence of parameter k than a restriction

Parameter d (max. variable domain size):

d is trivially polynomial-time checkable, thus also fpt

Parameter k (number of pahts in DTGs):

For each DTG, ask for the $k+1$ shortest paths and check if fewer than $k+1$ paths are returned.

Eppsteins algorithm (1998) can find the $k+1$ shortest paths in a DAG in $O(|E|+k+1)$ time, so checking parameter k is fpt.

Parameter w (treewidth of the causal graph):

A *tree decomposition* of a graph $G = \langle V, E \rangle$ is a tuple $\langle N, T \rangle$ where $N = \{N_1, \ldots, N_n\}$ is a family of subsets of V and T is a tree with nodes $N_1, \ldots, N_n$, satisfying the following properties:

1. Every $v \in V$ appears in at least one tree node.

2. For each $v \in V$, the set of nodes containing v form a connected subtree of T.

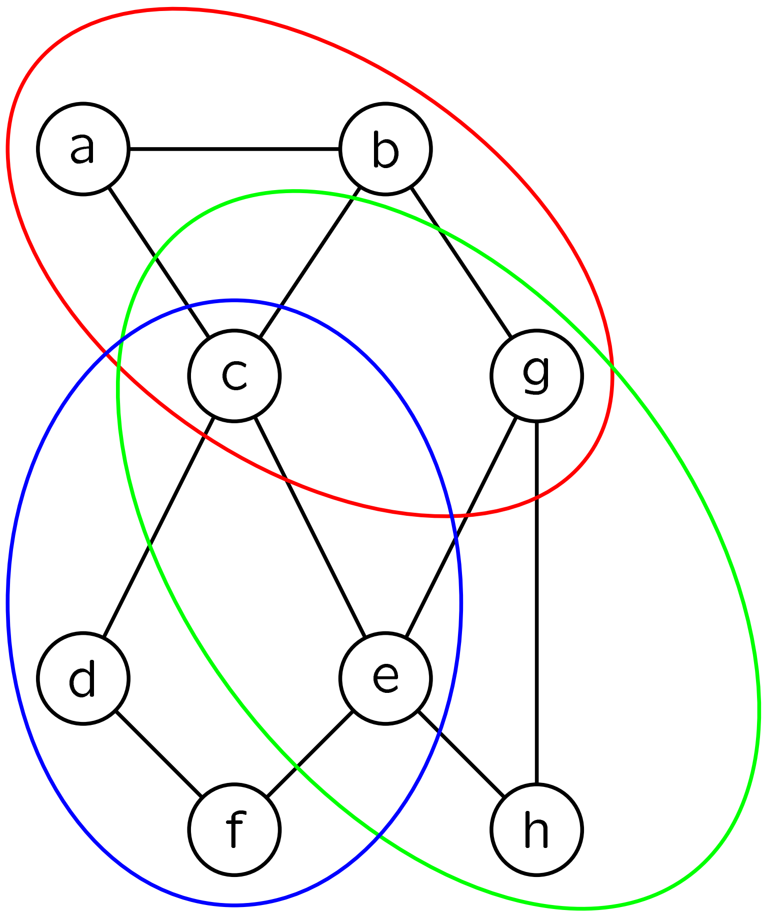
3. For every $\{u, v\} \in E$, there is some $N_i \in N$ such that $u, v \in N_i$, i.e. every pair of adjacent variables in G must appear together in some node.

The *width* of a tree decomposition is the size of its largest node minus one.

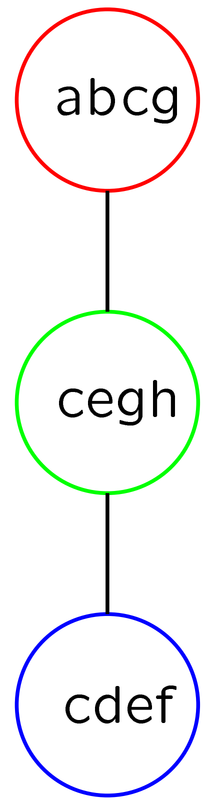
The *treewidth* of a graph G is the minimum width of all possible tree decompositions of G. (Perhaps not surprisingly, every tree has treewidth 1.)

Testing the treewidth of a graph is **NP**-complete but fpt in the treewidth

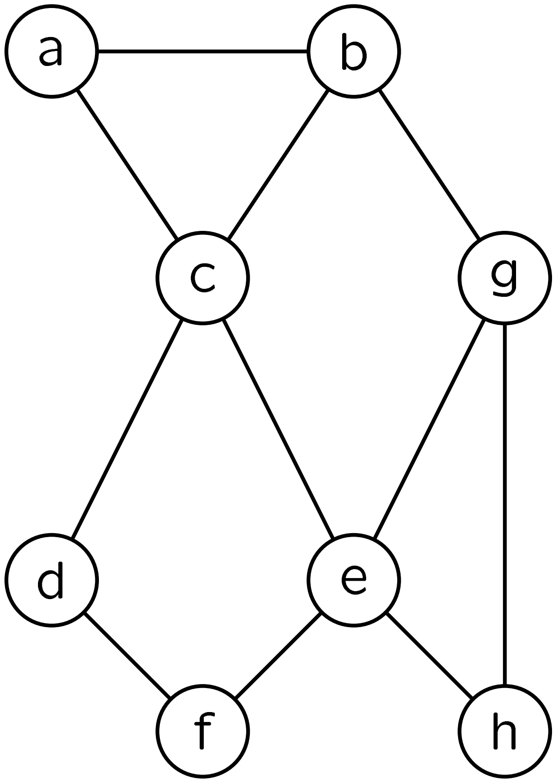
Parameter w is the treewidth of $U(CG(\mathbb{P}))$



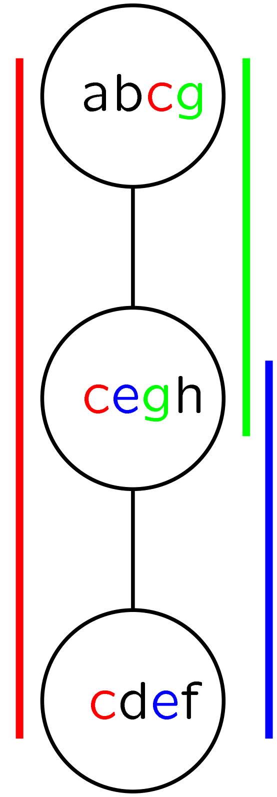
A graph



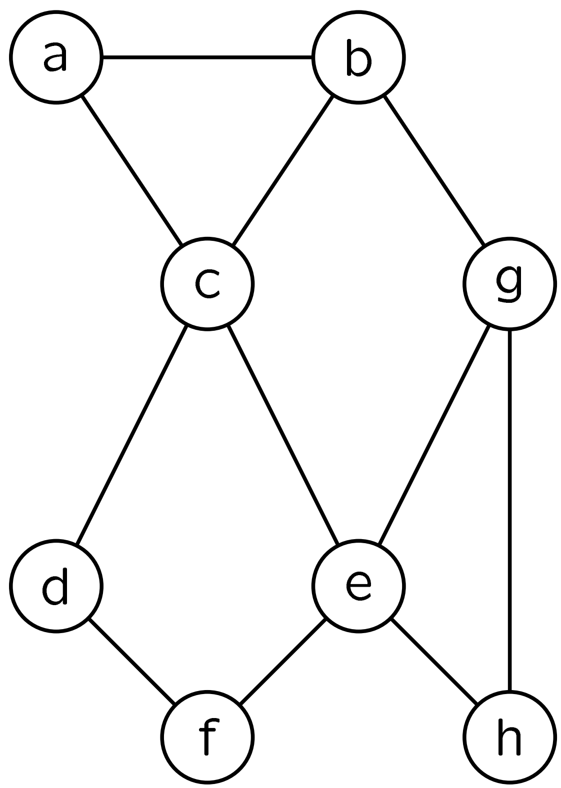
Tree decomposition
width = $(4 - 1) = 3$



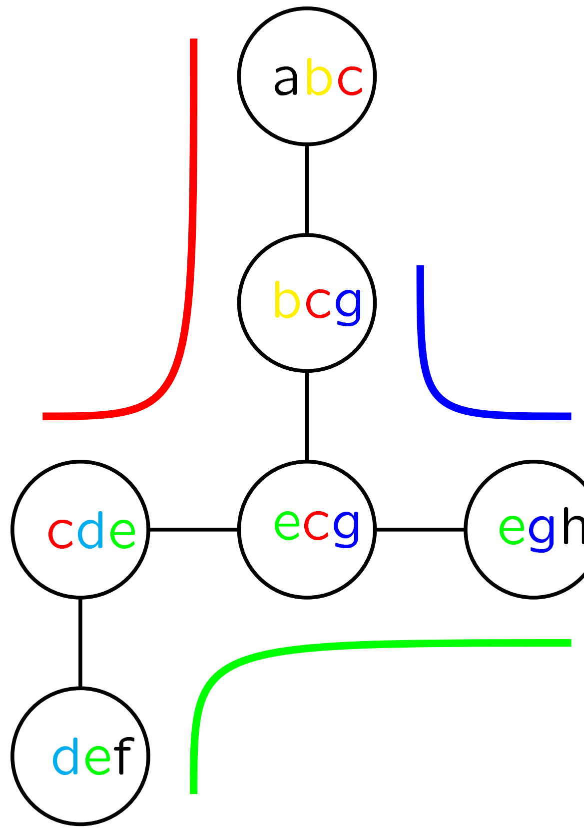
A graph



Tree decomposition
width = 3



A graph



Optimal tree decomposition
width = 2

Small nodes better than few nodes (usually)!

Main Theorem

Deciding if an instance $\mathbb{P}$ w. acyclic DTGs and arbitrary causal graph has a plan is fpt in parameters d, k, w.

Proof sketch

1. Construct a CSP instance $\mathbb{C}$ corresponding to $\mathbb{P}$

2. Prove that $\mathbb{C}$ solvable iff $\mathbb{P}$ solvable

3. Prove that constructing $\mathbb{C}$ and solving it is fpt in parameters $\langle d, k, w \rangle$

Constraint Satisfaction Problem (CSP):

A *binary* CSP instance $\mathbb{C} = \langle X, D, C \rangle$
- Set $X = \{x_1, \ldots, x_n\}$ of variables, each with finite domain $D(x_i)$
- Set C of constraints, i.e. relations of type $R_{i,j} \subseteq D(x_i) \times D(x_j)$

A *solution* for $\mathbb{C}$ is an assignment of variable values such that all relations in C are satisfied.

CSP Construction:

Let $\mathbb{P} = \langle V, D, A, s_I, s_G \rangle$ be an instance with acyclic DTGs.

Let $\langle N, T \rangle$ be a tree decomposition of $CG(\mathbb{P})$

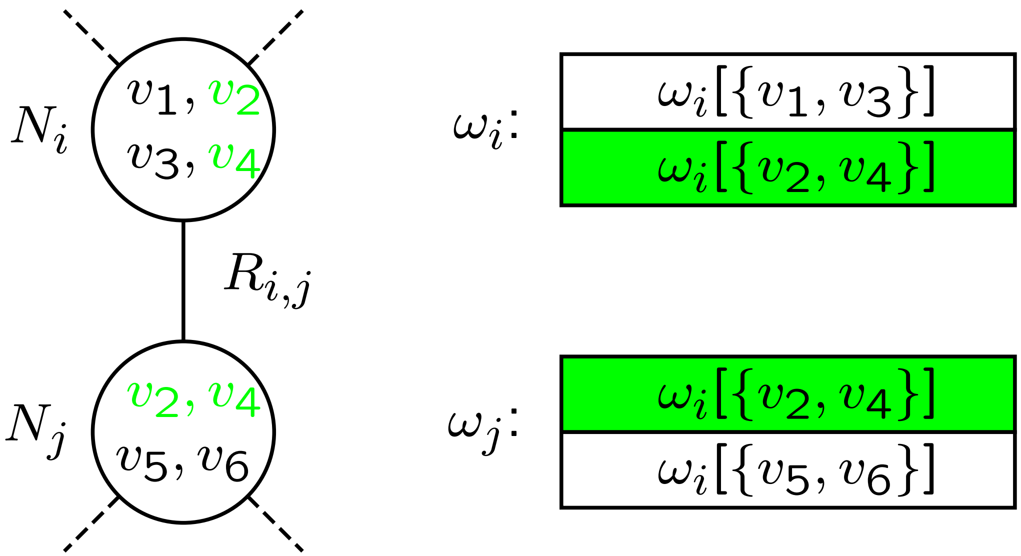
Define CSP instance $\mathbb{C} = \langle X, D, C \rangle$ as follows

- X contains one variable x_i for each node $N_i \in N$
- For each x_i, $D(x_i)$ is the set of plans for $\mathbb{P}[N_i]$
- For all adjacent N_i, N_j in T $(i < j)$,
  define $R_{i,j} \subseteq D(x_i) \times D(x_j)$ such that
  $R_{i,j}(\omega_i, \omega_j) \Leftrightarrow \omega_i[N_i \cap N_j] = \omega_j[N_i \cap N_j]$

That is, $R_{i,j}(\omega_i, \omega_j)$ holds if plans ω_i and ω_j agree for the common variables.

$\omega_i \in D(x_i)$ i.e. ω_i is a plan for $\mathbb{P}[N_i]$

$\omega_j \in D(x_j)$ i.e. ω_j is a plan for $\mathbb{P}[N_j]$



$$R_{i,j}(\omega_i, \omega_j) \Leftrightarrow \underline{\omega_i[\{v_2, v_4\}]} = \underline{\omega_j[\{v_2, v_4\}]}$$

Correctness Lemma:
$\mathbb{P}$ has a plan $\Leftrightarrow$ $\mathbb{C}$ has a solution

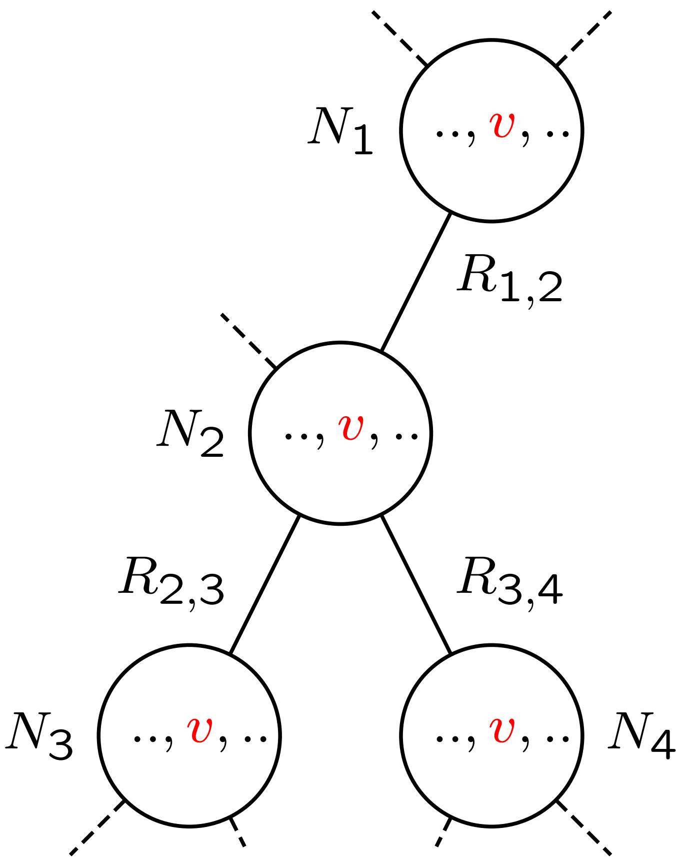
$\Rightarrow$:
Suppose ω is a plan for $\mathbb{P}$.
Suppose N_i, N_j adjacent nodes in T.
Let $\omega_i = \omega[N_i]$ and $\omega_j = \omega[N_j]$
Then trivially

$$\omega_i[N_i \cap N_j] = \omega[N_i \cap N_j] = \omega_j[N_i \cap N_j]$$

Hence $R_{i,j}(\omega_i, \omega_j)$ holds, so $\mathbb{C}$ is solvable.

$\Leftarrow$:
Suppose $\omega_1, \ldots, \omega_n$ is a solution to $\mathbb{C}$
We must prove that these plan fragments can be merged into one single plan for $\mathbb{P}$.

All nodes containing v form a subtree



If $\omega_1, \omega_2, \omega_3, \omega_4$ are part of a solution

then $R_{1,2}(\omega_1, \omega_2) \Rightarrow \omega_1[v] = \omega_2[v]$

$R_{2,3}(\omega_2, \omega_3) \Rightarrow \omega_2[v] = \omega_3[v]$

$R_{2,4}(\omega_2, \omega_4) \Rightarrow \omega_2[v] = \omega_4[v]$

so $\omega_1[v] = \omega_2[v] = \omega_3[v] = \omega_4[v]$

Hence, all node plans can be merged to one single plan for $\mathbb{P}$

Complexity Analysis:

1. Compute an optimal tree decomposition $\langle N, T \rangle$ of $CG(\mathbb{P})$. This is fpt in the treewidth w of G.

2. Suppose $N_i = \{v_1, \dots, v_m\}$

$$\begin{tikzcd}[column sep=small]
\mathrm{DTG}(v_1)\times \arrow[dr] & \dots & \times\mathrm{DTG}(v_m) \arrow[dl] \\
& TG(N_i) &
\end{tikzcd}$$

Each DTG has
$\leq k$ paths of length $\leq d$

$\Downarrow \ |N_i| \leq w + 1$

$TG(N_i)$ has $\leq k^{w+1}$ paths
of length $\leq d^{w+1}$

$\Downarrow \ |N_i| \leq w + 1$

$TG(N_i)$ has $\leq (kd)^{w+1}$ edges

Find the paths with Eppsteins algorithm in time
$O((kd)^{w+1} + k^{w+1}) = O((kd)^{w+1})$

3. Suppose $N_i = \{v_1, \ldots, v_m\}$

$$\begin{tikzcd}
TG(N_i) \arrow[dr] & & TG(N_j) \arrow[dl] \\
& R_{i,j} &
\end{tikzcd}$$

$TG(N_i), TG(N_j)$ have $\leq k^{w+1}$ paths

$$\Downarrow$$

$|D(x_i)|, |D(x_j)| \leq k^{w+1}$

$$\Downarrow$$

$|R_{i,j}| \leq (k^{w+1})^2 = k^{2(w+1)}$

Paths are of length $\leq d^{w+1}$

$$\Downarrow$$

Compare paths in time $O(d^{w+1})$

$$\Searrow \qquad\qquad\qquad\qquad \Swarrow$$

Compute $R_{i,j}$ in time $O((kd)^{2(w+1)} d^{w+1}) \subseteq O((kd)^{3(w+1)})$

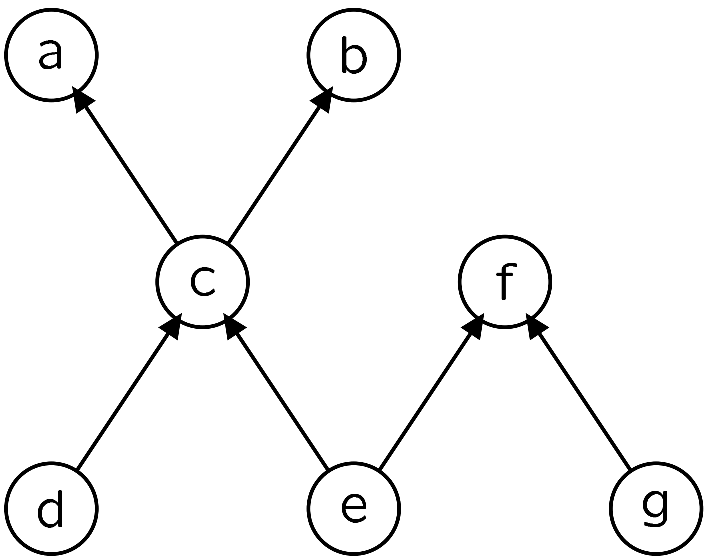
There are at most n nodes, so this construction takes time $O((kd)^{3(w+1)} \cdot n)$, which is fpt in $\langle d, k, w \rangle$

4. $\mathbb{C}$ is a binary CSP over a tree with $\leq n$ nodes where $D(x_i) \leq k^{w+1}$ for all x_i.
Using Decters and Pearls result (1989) for such CSPs we can solve $\mathbb{C}$ in time $O((k^{w+1})^2 \cdot n) = O(k^{w+1} \cdot n)$,
which is fpt in $\langle k, w \rangle$.

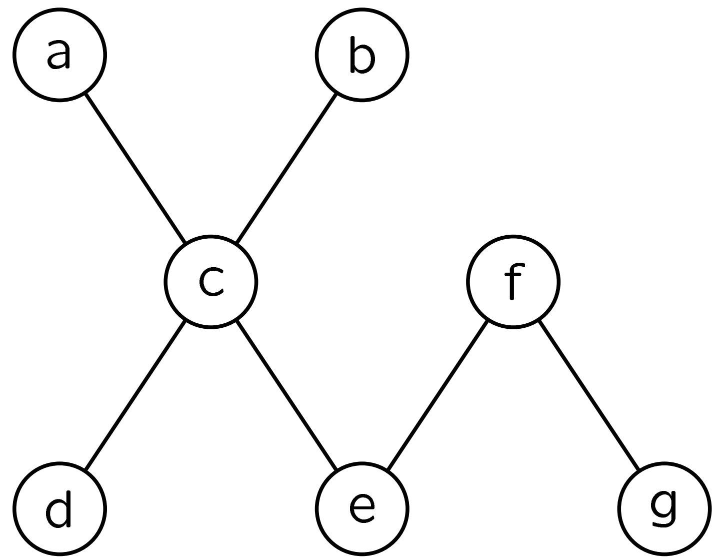
It follows that constructing and solving $\mathbb{C}$ is fpt in $\langle d, k, w \rangle$.

Polytree Causal Graphs

A digraph G is a *polytree* if its undirected version $U(G)$ is a tree



A polytree G
(Not a directed tree!)



$U(G)$
(a tree)

Corollary to Main Theorem:

Deciding if an instance $\mathbb{P}$ with acyclic DTGs has a plan is fpt in parameter k if $CG(\mathbb{P})$ is a polytree

Proof sketch:

Suppose $CG(\mathbb{P})$ is a polytree. Then $w = 1$.

Above, d always appears in the form d^{w+1}
This becomes d^2 when $w = 1$.
The dominating expression is $O((kd)^{3(w+1)}) = O(k^6 d^6 n))$.

However, $d \leq n$ so we get $O(k^6 \cdot n^7)$ which is fpt in k alone.

Optimisation

CSOP is the problem of finding the minimum weight solution for a weighted CSP instance.

Färnqvists corollary (2012): Solving CSOP is fpt in the parameters

- max. domain size and
- treewidth of the constraint graph.

Can be used to prove that the two previous fpt results for planning hold also for finding an *optimal* plan.

1. Define the weight of a path as its length.

2. Let $\omega_1, \ldots, \omega_n$ be a solution to $\mathbb{C}$.
Each action a that has an effect on variable v will appear in ω_i for each N_i s.t. $v \in N_i$.
Arbitrarily associate a with exactly one such N_i and count a only when it occurs in ω_i.
This will give the correct count for the final plan for $\mathbb{P}$.

3. Using the previous proof we get that also finding a shortest plan for $\mathbb{P}$ is fpt in $\langle d, k, w \rangle$.

Delete Relaxation Heuristics

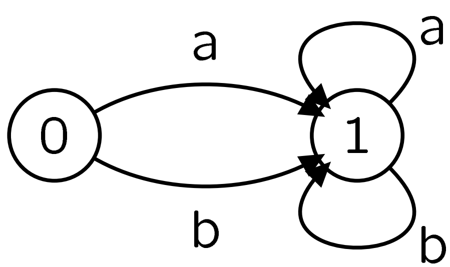
Acyclic DTGs may seem very restricted
However, the results can easily be applied to compute the *delete relaxation heuristic* h^+

h^+: Length of shortest plan when negative effects of actions are ignored

DTGs have loops, but each action occurs at most once in a shortest plan
Hence, the number of relevant paths is bounded

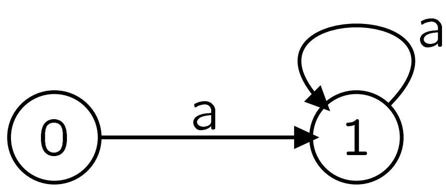
a: $\{\} \Rightarrow \{x = 1, y = 1\}$

b: $\{\} \Rightarrow \{x = 1, z = 1\}$

DTG(x): 

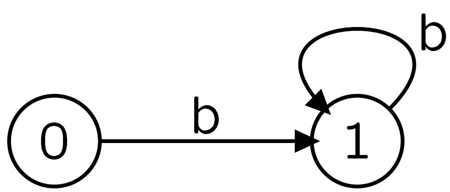
a, b, ab, ba possible (4 paths)

aa, aab, abb, bba,... irrelevant

DTG(y): 

a possible (1 path)

aa, aa, aaa,... irrelevant

DTG(z): 

b possible (1 path)

bb,bbb,... irrelevant

Conclusions

Deciding if there is a plan, and even finding an optimal plan, is fpt in $\langle d, k, w \rangle$, when all DTGs are acyclic.

Deciding if there is a plan, and even finding an optimal plan, is fpt in k, when all DTGs are acyclic and the causal graph is a polytree.

The results imply that computing the h^+ heuristic is fpt in $\langle d, k, w \rangle$, and fpt in k if the causal graph is a polytree

All parameters are fpt testable

Future Work

Study other types of restricted causal graphs than polytrees

Non-acyclic DTGs. Many ways forward, for instance:

• Identify SCCs and let k denote # of paths in condensed DTG. Add parameters for the SCCs.

• Allow only very restricted forms of acyclicity, with appropriate parameters

• Parameterize the number of turns in loops and unroll the loops