# A Refined Understanding of Cost-optimal Planning with Polytree Causal Graphs*

**Christer Bäckström**[1†], **Peter Jonsson**[1] and **Sebastian Ordyniak**[2]
[1]Linköping University
[2]University of Sheffield
{christer.backstrom, peter.jonsson}@liu.se, sordyniak@gmail.com

## Abstract

Complexity analysis based on the causal graphs of planning instances is a highly important research area. In particular, tractability results have led to new methods for constructing domain-independent heuristics. Important early examples of such results were presented by, for instance, Brafman & Domshlak and Katz & Keyder. More general results based on polytrees and bounding certain parameters were subsequently derived by Aghighi et al. and Ståhlberg. We continue this line of research by analyzing cost-optimal planning for instances with a polytree causal graph, bounded domain size and bounded depth. We show that no further restrictions are necessary for tractability, thus generalizing the previous results. Our approach is based on a novel method of closely analysing optimal plans: we recursively decompose the causal graph in a way that allows for bounding the number of variable changes as a function of the depth, using a reording argument and a comparison with prefix trees of known size. We then transform the planning instances into tree-structured constraint satisfaction instances.

## 1 Introduction

A common approach for identifying tractable fragments of planning is to analyse the *causal graph* [Knoblock, 1994], a directed graph where the vertices represent variables and the arcs represent certain dependencies between variables. By combining restrictions on the structure of this graph with bounds on certain problem-specific parameters, many different tractability results have been obtained. It was early noted that if the causal graph is acyclic, then all actions are unary, i.e. change one variable only, yet such restricted instances can be useful even in practice [Williams and Nayak, 1997]. Helmert [2006] pioneered the idea of defining heuristics based on subgraphs of the causal graph. He removed arcs in the graph to be able to find acyclic subgraphs, which made it easier to define heuristics. However, not even acyclicity is sufficient in the general case; planning is still **PSPACE**-complete even when restricted to arbitrary acyclic causal graphs [Jonsson *et al.*, 2014]. Hence, there has been much focus on various restricted types of acyclic graphs, for example *forks*, *inverted forks* and *hourglasses*. Cost-optimal planning is **NP**-hard if the causal graph is of either of these types with no further restrictions, but becomes tractable for all three types if we also bound the variable domain size by a constant [Katz and Keyder, 2012; Katz and Domshlak, 2010].

A directed graph is a *polytree* if it is acyclic and its underlying undirected graph is a tree. Problems with causal graphs that are polytrees have been intensively studied in the literature. It is easy to verify that (inverted) forks and hourglasses are polytrees. Aghighi *et al.* [2015] show that cost-optimal planning is tractable for instances with bounded *domain size* and a polytree causal graph with bounded *diameter*, the length of the longest path in the underlying undirected graph. Another popular parameter is the *in-degree* of the causal graph, i.e. the maximum number of arcs that go into a vertex. The in-degree is 1 for forks but unbounded for inverted forks and hourglasses. Cost-optimal planning is tractable for instances with polytree causal graphs, domain size 2 and bounded in-degree [Katz and Domshlak, 2008]. This result cannot be generalised to arbitrarily large domains: even satisficing planning is NP-hard for domain size 5 and in-degree 1 [Giménez and Jonsson, 2009]. Ståhlberg [2017] considered the *depth* of the causal graph, i.e. the length of the longest directed path (which is obviously upper bounded by the diameter). He showed that cost-optimal planning is tractable for instances with bounded domain size and polytree causal graphs with bounded depth and in-degree. We improve on his result, showing that it is sufficient to bound only the depth and the domain size to achieve tractability. The resulting tractable fragment is maximal for polytrees in the sense that we cannot drop the domain size bound nor the depth bound with retained tractability; it is known that cost-optimal planning is **NP**-hard for causal graphs of arbitrary depth, even if the domain size is 2 [Giménez and Jonsson, 2008] and for inverted-fork causal graphs and unbounded domain size [Domshlak and Dinitz, 2001].

The structure of this extended abstract follows the original

*This is an extended abstract of the paper “A Refined Understanding of Cost-optimal Planning with Polytree Causal Graphs” which won the best-paper award at the SoCS-2018 conference.

†Contact Author

full paper [Bäckström *et al.*, 2018], using the same sections and the same numbering of results.

## 2 Preliminaries

A SAS$^+$ *planning instance* is a tuple $\mathbb{P} = \langle V, A, s_I, s_G, c\rangle$ where $V$ is a set of variables, with an implicit domain $D$, $A$ is a set of *actions* and $c : A \to \mathbb{Q}_{\geq 0}$ is a *cost function*. The *initial state* $s_I$ is a total state and the *goal* $s_G$ is a partial state. Each action $a \in A$ has two partial states the *precondition* $\text{pre}(a)$ and the *effect* $\text{eff}(a)$. An action sequence $\omega$ is a plan for $\mathbb{P}$ if transforms $s_I$ to a state $s$ that satisfies $s_G$. The *length* $|\omega|$ of $\omega$ is the number of actions in $\omega$ and the *cost* $c(\omega)$ is the total cost of the actions in $\omega$. We also define $C(v, \omega)$ as the number of value changes of $v$ when executing $\omega$. Furthermore, $\omega$ is a *cost-optimal plan* for $\mathbb{P}$ if there is no plan $\omega'$ for $\mathbb{P}$ such that $c(\omega') < c(\omega)$; and $\omega$ is a *shortest cost-optimal plan* for $\mathbb{P}$ if it is cost optimal and there is no plan $\omega'$ for $\mathbb{P}$ such that $c(\omega') = c(\omega)$ and $|\omega'| < |\omega|$. The latter concept is important in the presence of zero-cost actions, since a cost-optimal plan can then be arbitrarily long. In order to find a cost-optimal plan, it is obviously sufficient to find a shortest cost-optimal plan. The projection of $\mathbb{P}$ to a subset $V' \subseteq V$ is denoted $\mathbb{P}[V']$ and is identical to $\mathbb{P}$ except that all components are restricted to the variables in $V'$. The projection $\omega[V']$ of a plan $\omega$ to a set $V' \subseteq V$ of variables is the subsequence of actions in $\omega$ with an effect on at least one variable in $V'$.

The *causal graph* $CG(\mathbb{P})$ for $\mathbb{P}$ describes how the variables depend on each other, as implicitly defined by the actions. It is defined as the directed graph $CG(\mathbb{P}) = \langle V, E\rangle$ where for all distinct $v, w \in V$, $\langle v, w\rangle \in E$ if there is some action that either has a precondition on $v$ and an effect on $w$ or it has effects on both $v$ and $w$. An action is *unary* if it has an effect on exactly one variable, and it is immediate that all actions must be unary if $CG(\mathbb{P})$ is acyclic, and the following proposition is immediate since a shortest (cost-optimal) plan cannot have any redundant actions.

**Proposition 1.** *Let $\mathbb{P}$ be a unary SAS$^+$ instance. If $\omega$ is a shortest or shortest cost-optimal plan for $\mathbb{P}$, then $|\omega[v]| = C(v, \omega)$ for all $v \in V$.*

A directed graph $G = \langle V, E\rangle$ is a *polytree* if it is acyclic and the undirected variant of it is a tree, i.e. if we ignore the direction of the edges then $G$ must be connected and contain no cycles. The *depth* $d(v)$ of a vertex $v \in V$ is the length of the longest directed path from $v$ to any sink in $G$.

## 3 Planning for Polytrees

We focus on planning for instances where the causal graph is a polytree, first deriving a bound on the complexity of cost-optimal planning for such instances given that we know how many variable changes we must consider. Then we present such a bound as a function $B(s, d)$ of the domain size and the depth, such that for every variable $v$ and every shortest cost-optimal plan $\omega$, it holds that $C(v, \omega) \leq B(s, d(v))$. Finally, we combine this into our main result (Theorem 4).

### 3.1 Planning as CSP

We first improve a known complexity result.

**Lemma 2.** *Let $\mathbb{P} = \langle V, A, s_I, s_G, c\rangle$ be a SAS$^+$ instance such that $CG(\mathbb{P})$ is a polytree. Let $B$ be an upper bound on $C(v, \omega)$ for all $v \in V$ and all shortest cost-optimal plans $\omega$ for $\mathbb{P}$. Then cost-optimal planning can be solved in time $O((Bs^{B+1})^6 n)$, where $n = \|\mathbb{P}\|$ is the instance size.*

*Proof.* If we know an upper bound $k$ on the number of walks that has to be considered in any domain-transition graph (DTG), then plan satisfiability can be solved in time $O((ks)^6 n)$ Bäckström [2014]. This result uses a CSP encoding based on a tree decomposition of the causal graph, exploiting that CSP can be solved in time $O(N_{\mathbb{C}} D_{\mathbb{C}}^2)$ for tree primal graphs [Dechter and Pearl, 1989], where $N_{\mathbb{C}}$ is the number of CSP variables and $D_{\mathbb{C}}$ their domain size. Bäckström [2014] also shows how to solve cost-optimal planning this way, but no explicit complexity figure is given. However, since also cost-optimal CSP can be solved in time $O(N_{\mathbb{C}} D_{\mathbb{C}}^2)$ [Cooper and Schiex, 2004, Theorem 5.4] for tree primal graphs, it follows from the proof of Theorem 12 in Bäckström [2014] that also cost-optimal planning can be solved in time $O((ks)^6 n)$. Let $B$ be an upper bound on the length of the subplan for any variable in a shortest cost-optimal plan. Then there are at most $\sum_{i=0}^{B} (s-1)^i \leq Bs^B$ walks of length $B$ in any DTG, so it follows that we can find a shortest cost-optimal plan in time $O((Bs^{B+1})^6 n)$ □

### 3.2 Bounds for Polytree Causal Graphs

In the derivation and definition of the bound function $B$, we need a function $\tau$, defined such that $\tau(s, h)$ is the number of nodes in a maximal tree of height $h$ where the root has branching factor $s$ and all other interior nodes have branching factor $s - 1$. The reason for this definition will become clear later. For all $s \geq 2$ and $h \geq 1$, we thus have

$$\tau(s, h) = 1 + s \sum_{i=0}^{h-1} (s-1)^i$$

which can be bounded from above as $\tau(s, h) \leq 3(s-1)^h$. We can now define the bound function $B$ of the domain size $s$ and the depth $d$ as

$$B(s, d) = \begin{cases} s - 1, & \text{if } d = 0 \\ \tau\big(s, B(s, d-1) + 1\big) + (s - 2), & \text{if } d > 0. \end{cases}$$

We will show that this function is an upper bound on the number of variable changes in all shortest cost-optimal plans, and it can itself be upper bounded by a 'tower function' with $d$ levels of exponentiation of the form

$$4(s-1)^{4(s-1)^{\cdot^{\cdot^{\cdot 4(s-1)^s}}}}.$$

**Lemma 3.** *Let $\mathbb{P} = \langle V, A, s_I, s_G, c\rangle$ be a SAS$^+$ instance with a polytree causal graph. If $\omega$ is a shortest cost-optimal plan for $\mathbb{P}$, then $C(v, \omega) \leq B(s, d(v))$ for all $v \in V$.*

We will sketch the proof of this lemma in Sec. 4, and can now state our main result.

**Theorem 4 (Main result).** *Let $\mathbb{P} = \langle V, A, s_I, s_G, c\rangle$ be a SAS$^+$ instance such that $CG(\mathbb{P})$ is a polytree with maximum depth $d$. If $\mathbb{P}$ is solvable, we can find a cost-optimal plan for it in time $O\big((B(s, d) \cdot s^{B(s,d)+1})^6 \cdot \|\mathbb{P}\|\big)$.*

*Proof.* It is sufficient to find a shortest cost-optimal plan, so the result follows from Lemmas 2 and 3. □

## 4 Plan Lengths for Polytree Causal Graphs

In this section we will informally sketch the proof of Lemma 3 using examples (we refer the reader to the original publication [Bäckström *et al.*, 2018] for the full formal proof). The proof is by induction over the depth of vertices (i.e. variables) in the causal graph. Figure 1 shows an example of a polytree which is drawn such that the vertices are aligned according to depth. We will use this polytree as an example for illustrating the induction.

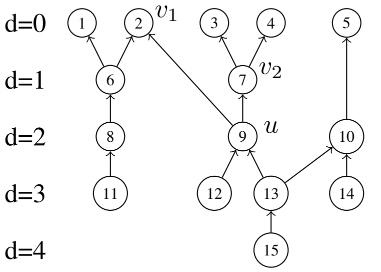

Figure 1: An example polytree causal graph.

The base case consists of all variables at depth 0 (1–5 in the example). Since a variable $v$ at depth 0 has no outgoing arcs, no other variables depend on it. Hence, the subplan $\omega[v]$ for this variable cannot contain any cycles, since $\omega$ is a shortest cost-optimal plan. Thus, in the worst case $\omega[v]$ is a Hamilton cycle in the DTG for $v$, i.e. of length $s - 1$. It follows that $C(v, \omega) \leq s - 1 = B(s, 0)$ for all $v$ such that $d(v) = 0$.

In the induction step, we must prove that if $C(v, \omega) \leq B(s, d(v))$ for all $v \in V$ with $d(v) \leq d$, for some $d$, then it also holds that $C(v, \omega) \leq B(s, d(v))$ for all $v \in V$ with $d(v) = d+1$. We will explain the induction step by example, assuming the claim holds up to depth 1, i.e. we want to prove the claim for depth 2. We assume an arbitrary variable $u$ at depth $d + 1$ and let $v_1, \ldots, v_m$ denote the incident variables of its outgoing arcs. As an example, we choose $u = 9$ as an arbitrary variable at this depth, so we get $v_1 = 2$ and $v_2 = 7$. At least one of $v_1, \ldots, v_m$ must be at depth $d$, while the others can be at any depth $\leq d$. In the example we have $v_2$ at depth 1 but $v_1$ at depth 0. If we were to remove the outgoing arcs from $u$, we would partition the graph into three sets $V_1$, $V_2$ and $U$ such that $v_1 \in V_1$, $v_2 \in V_2$ and $u \in U$. This partitioning is illustrated in Figure 2.

The remainder of this proof sketch is divided into three steps as in the original full proof.

**Step I.** Let $\chi_1$ be the sequence of defined preconditions on $u$ in the actions in subplan $\omega[v_1]$ and define $\chi_2$ analogously for $v_2$. Also let $\psi$ be the sequence of values that $u$ passes, from the initial value to the goal value. We define a release-time function $r$ that maps the indices of $\chi_1$ and $\chi_2$ to indices of $\psi$, witnessing that $\chi_1$ and $\chi_2$ are subsequences of $\psi$. We write $r_1$ and $r_2$ for the restrictions of $r$ to $\chi_1$ and $\chi_2$ respectively. We further say that $r_1$ is minimal if every index in $\chi_1$ is mapped as early as possible in $\chi$. Note that if there exists a release-time function for $\chi_1$, then there must also exist a unique minimal one. Figure 3 shows two different release-time functions, $r_1$ and $r_1'$, for $\chi_1$, where $r_1'$, but not $r_1$, is minimal. Furthermore, the function $r$ is minimal if both $r_1$ and $r_2$ are minimal.

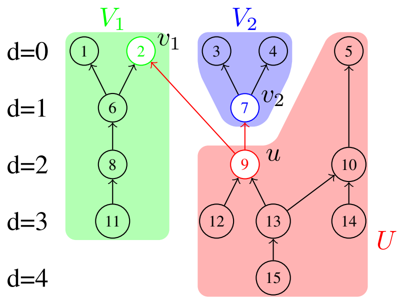

Figure 2: Variable partition of the polytree causal graph.

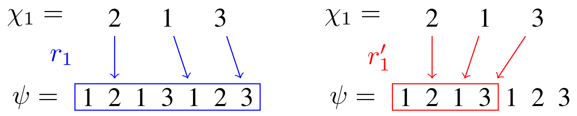

Figure 3: Example release-time functions.

We further note that the minimal release-time function always maps each sequence $\chi_i$ to the shortest possible prefix of $\psi$. In Figure 3 we see that $r_1'$ maps $\chi_1$ to a shorter prefix than $r_1$ does. Furthermore, if two consecutive elements in $\chi_1$ have the same value, then the minimal release-time function must map these to the same index in $\psi$. Hence, we may assume that $\chi_1$ is an alternating sequence in the worst case. Clearly, $\chi_1$ and $\chi_2$ must be subsequences of $\psi$ in this case.

We note that the partition of $V$ into the sets $V_1, V_2$ and $U$ also implies a corresponding partition of the plan $\omega$ into three subplans $\omega[V_1]$, $\omega[V_2]$ and $\omega[U]$, where every action occurence in $\omega$ belongs to exactly one of these three subplans since all actions are unary. In Figure 2 we see that the only arc from $U$ to $V_1$ is from $u$ to $v_1$, so the only variable in $V_1$ that depends on some variable in $U$ is $v_1$ and it depends only on $u$. Analogously, only $v_2$ in $V_2$ depends on $u$. Furthermore, no variable in $V_1$ depend on any variable in $V_2$ and vice versa. It follows that the subplans $\omega[V_1]$ and $\omega[V_2]$ are independent of each other, but both depend on the subplan $\omega[U]$. However, since there are only arcs from $u$ to $v_1$ and $v_2$, we only need to syncrhonise the subplans $\omega[v_1]$ and $\omega[v_2]$ with the subplan $\omega[u]$, which will be exploited later in the proof. Since $\omega[V_1]$ and $\omega[V_2]$ do not depend on each other, the corresponding release-time functions $r_1$ and $r_2$ are independent of each other, i.e. we can choose subsequences in $\psi$ for $\chi_1$ and $\chi_2$ independently of each other. It follows that it is always possible to reorder $\omega$ into a plan $\omega'$ that contains exactly the same action occurences as $\omega$ but where the release-time function for $\chi_1$ and $\chi_2$ is minimal. This is illustrated in Figure 4. We first split $\omega$ into $\omega[V_1]$, $\omega[V_2]$ and $\omega[U]$. We retain the internal order of each of these three subplans, but reorder them with respect to each other. More precisely, we can interleave these subplans freely as long as we syncrhonise $\omega[v_1]$ with

$\omega[u]$ and $\omega[v_2]$ with $\omega[u]$. In particular, we can interleave them such that the release time functions $r_1$ and $r_2$ are both minimal. The point of this reordering is that $\omega'$ is a permutation of $\omega$, so it has the same length and cost as $\omega$, but it is easier to analyse since we know that the release-time functions are minimal.

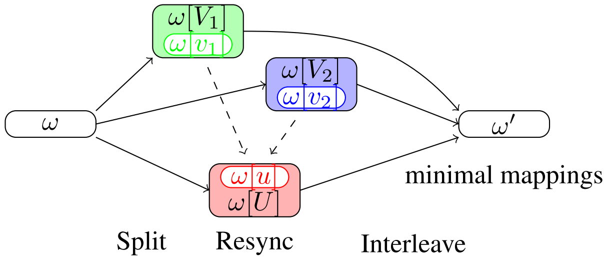

Figure 4: Plan reordering.

**Step II.** For this step we need a slightly more complex example, so assume that variable $u$ has four outgoing arcs to variables $v_1, \ldots, v_4$, with corresponding sequences $\chi_1, \ldots, \chi_4$ of preconditions on $u$. Also suppose that $\chi_1 = 131$, $\chi_2 = 132$, $\chi_3 = 312$, $\chi_4 = 323$ and $\psi = 123123$. We can then build a prefix tree $T$ that contains a node for every prefix of these four sequences, and then define a minimal tree mapping $r_T$ from the nodes of $T$ to $\psi$ as follows. The root of the prefix tree is the empyt prefix $\epsilon$, which we map to 0, i.e. $r_T(\epsilon) = 0$. For all other nodes $n$, if $n = x_1 \ldots x_m$ is a child of $n'$ in $T$, then $n' = x_1 \ldots x_{m-1}$, so we let $r_T(n)$ be the smallest index such that $r_T(n') < r_T(n)$ and position $r_T(n)$ in $\psi$ has value $x_m$. That is, $r_T$ is a witness for the earliest occurence in $\psi$ of each prefix in $T$. This minimal tree mapping is unique. Note that we can always reconstruct the unique minimal release-time function from this minimal tree mapping. Furthermore, let $T_{max}$ be the maximal prefix tree with the same depth as $T$, i.e. it contains all possible alternating prefixes of this length. Then $T$ must be a subtree of $T_{max}$. Figure 5 shows $T_{max}$ in green overlayed by the subtree $T$ in thick black lines, and the unique minimal tree mapping from $T$ to $\psi$ is shown with blue dashed arrows.

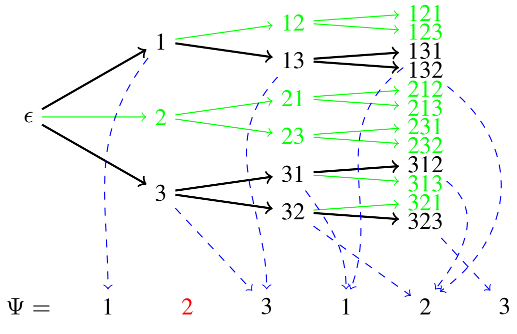

Figure 5: Tree embedding.

**Step III.** Let $r_T$ be the unique minimal tree mapping from $T$ to $\psi$. In Figure 5, $r_T$ does not map any node in $T$ to the second index in $\psi$, marked in red. Since we do not know how many such unmapped indices may exist in general, we cannot use the size of $T$ to bound the length of $\psi$. However, we can use $T_{max}$ to provide such a bound. We construct a partial tree mapping $r'_T$ from $T_{max}$ to $\psi$ such that $r'_T$ is minimal and extends $r_T$. For each index $i$ in $\psi$ which is not mapped by $r_T$, we can always find at least one unique node $n$ in $T_{max}$ which is not in $T$ such that if $r'_T$ is defined for $n$, then $r'_T(n) = i$ must hold. In the example we can choose either $n = 2$ or $n = 12$ as the node to map to index 2. Hence, the number of indices in $\psi$ that are not mapped by $R$ are at most as many as the number of nodes in $T_{max}$ that are not in $T$. Since $T$ is a subtree of $T_{max}$, the length of $\psi$ is bounded by the size of $T_{max}$ (plus $\leq s - 1$ actions for achieving the goal). If $T_{max}$ has height $h$, the size of $T_{max}$ is $\tau(s, h)$ so $|\psi| \leq \tau(s, h + 1) + (s - 1)$. We thus get that $C(u, \omega) = |\psi| - 1 \leq \tau(s, h+1) + (s-1) - 1 = \tau(s, h+1) + s - 2 \leq \tau(s, B(s, d) + 1) + s - 2 = B(s, d + 1)$ which ends the induction.

## 5 Concluding Remarks

We have presented a polynomial-time algorithm for cost-optimal planning restricted to polytree causal graphs, bounded depth, and bounded domain size. It, thus, advances the tractability frontier since previous tractability results using these restrictions also require further restrictions. Our algorithm is based on transforming cost-optimal planning into VCSP, using a transformation originally suggested by Bäckström [2014]. The main technical result is a bound on the number of variable changes defined in terms of the depth of the causal graph and the domain size. This result is maximal in the sense that dropping the domain size bound or the depth bound size leads to an **NP**-hard problem. A natural open question is thus how to generalise the tractability result to larger classes of planning instances. It is not obvious what other parameter bounds would lead to tractability so it may be easier to consider larger classes of causal graphs than polytrees. The obvious generalisation is to consider causal graphs that have bounded tree-width—this is a method that have led to a large number of tractability results in many different areas of computer science, with a few examples also in planning [Brafman and Domshlak, 2006; Domshlak and Nazarenko, 2013; Bäckström, 2014]. The very same planning algorithm would be useful also in this case: it runs in polynomial time whenever the tree-width of the causal graph and the number of variable changes are bounded. Unfortunately, it is not at all clear how to generalise the bound on variable changes when the causal graph has bounded tree-width larger than one. Although our algorithm is polynomial-time, its running time is admittedly not impressive. However, previous tractability results like Aghighi *et al.* [2015] and Ståhlberg [2017], have similar tower functions despite using more restrictions than we do. We believe that our bounds can be considerably improved by an even more careful analysis, combining these proof techniques with others to give an even more refined picture of optimal plans.

## Acknowledgments

Bäckström is partially supported by the Swedish Research Council (VR) under grant 621-2014-4086.

## References

[Aghighi *et al.*, 2015] Meysam Aghighi, Peter Jonsson, and Simon Ståhlberg. Tractable cost-optimal planning over restricted polytree causal graphs. In *Proc. 29th AAAI Conference on Artificial Intelligence (AAAI 2015), Austin, TX, USA*, pages 3225–3231, 2015.

[Bäckström *et al.*, 2018] Christer Bäckström, Peter Jonsson, and Sebastian Ordyniak. A refined understanding of cost-optimal planning with polytree causal graphs. In *Proc. 11th Int'l Symp. Combinatorial Search, SOCS 2018, Stockholm, Sweden, 2018*, pages 19–27, 2018.

[Bäckström, 2014] Christer Bäckström. Parameterising the complexity of planning by the number of paths in the domain-transition graphs. In *Proc. 21st Eur. Conf. Artif. Intell. (ECAI-14), Prague, Czech Rep.*, pages 33–38, 2014.

[Brafman and Domshlak, 2006] Ronen I. Brafman and Carmel Domshlak. Factored planning: How, when, and when not. In *Proc. 21st Nat'l Conf. Artif. Intell. 2006, Boston, MA, USA*, pages 809–814, 2006.

[Cooper and Schiex, 2004] Martin C. Cooper and Thomas Schiex. Arc consistency for soft constraints. *Artif. Intell.*, 154(1-2):199–227, 2004.

[Dechter and Pearl, 1989] Rina Dechter and Judea Pearl. Tree clustering for constraint networks. *Artif. Intell.*, 38(3):353–366, 1989.

[Domshlak and Dinitz, 2001] Carmel Domshlak and Yefim Dinitz. Multi-agent off-line coordination: Structure and complexity. In *Proc. 6th European Conference on Planning (ECP 2001), Toledo, Spain*, pages 34–43, 2001.

[Domshlak and Nazarenko, 2013] Carmel Domshlak and Anton Nazarenko. The complexity of optimal monotonic planning: The bad, the good, and the causal graph. *J. Artif. Intell. Res.*, 48:783–812, 2013.

[Giménez and Jonsson, 2008] Omer Giménez and Anders Jonsson. The complexity of planning problems with simple causal graphs. *J. Artif. Intell. Res.*, 31:319–351, 2008.

[Giménez and Jonsson, 2009] Omer Giménez and Anders Jonsson. Planning over chain causal graphs for variables with domains of size 5 is NP-hard. *J. Artif. Intell. Res.*, 34:675–706, 2009.

[Helmert, 2006] Malte Helmert. The fast downward planning system. *J. Artif. Intell. Res.*, 26:191–246, 2006.

[Jonsson *et al.*, 2014] Anders Jonsson, Peter Jonsson, and Tomas Lööw. Limitations of acyclic causal graphs for planning. *Artif. Intell.*, 210:36–55, 2014.

[Katz and Domshlak, 2008] Michael Katz and Carmel Domshlak. New islands of tractability of cost-optimal planning. *J. Artif. Intell. Res.*, 32:203–288, 2008.

[Katz and Domshlak, 2010] Michael Katz and Carmel Domshlak. Implicit abstraction heuristics. *J. Artif. Intell. Res.*, 39:51–126, 2010.

[Katz and Keyder, 2012] Michael Katz and Emil Keyder. Structural patterns beyond forks: Extending the complexity boundaries of classical planning. In *Proc. 26th AAAI Conf. Artif. Intell. (AAAI 2012), Toronto, ON, Canada*, 2012.

[Knoblock, 1994] Craig A. Knoblock. Automatically generating abstractions for planning. *Artif. Intell.*, 68(2):243–302, 1994.

[Ståhlberg, 2017] Simon Ståhlberg. *Methods for Detecting Unsolvable Planning Instances using Variable Projection*. PhD thesis, Linköping University, Sweden, 2017.

[Williams and Nayak, 1997] Brian C. Williams and P. Pandurang Nayak. A reactive planner for a model-based executive. In *Proc. 15th International Joint Conference on Artificial Intelligence (IJCAI 1997), Nagoya, Japan*, pages 1178–1185, 1997.