The Eleventh International Symposium on Combinatorial Search
(SoCS 2018)

A Refined Understanding of Cost-Optimal Planning with Polytree Causal Graphs

Christer Bäckström
Linköping University
christer.backstrom@liu.se

Peter Jonsson
Linköping University
peter.jonsson@liu.se

Sebastian Ordyniak
University of Sheffield
sordyniak@gmail.com

Abstract

Complexity analysis based on the causal graphs of planning instances has emerged as a highly important area of research. In particular, tractability results have led to new methods for the identification of domain-independent heuristics. Important early examples of such tractability results have been 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 restricted to instances with a polytree causal graph, bounded domain size and bounded depth (i.e. the length of the longest directed path in the causal graph). 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 can then transform the planning instances into constraint satisfaction instances; an idea that has previously been exploited by, for example, Brafman & Domshlak and Bäckström. This allows us to utilise efficient algorithms for constraint optimisation over tree-structured instances.

1 Introduction

Analysing the complexity of planning has been an important research area for quite some time, and identifying tractable (i.e. polynomial-time) fragments of planning is very important. An obvious application is efficient planning in concrete systems, like control systems for industrial plants (Cooper, Maris, and Régnier 2014) and spacecrafts (Muscettola et al. 1998). Another highly important usage is in the construction of efficient search heuristics based on solving tractable fragments. This is a well-studied topic so we will not go into details here; the interested reader is referred to, for example, Helmert (2004), Helmert, Haslum, and Hoffmann (2007), and Katz and Domshlak (2010). Further motivations are planning problem decomposition (aka. *factored planning*) (Brafman and Domshlak 2006) and multi-agent planning (Brafman and Domshlak 2013).

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, Jonsson, and Lööw 2014). Hence, there has been much focus on various restricted types of acyclic graphs, for example *forks*, *inverted forks* and *hourglasses*, which are illustrated in Figure 1. 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 (an example appears in Fig. 2). 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, Jonsson, and Ståhlberg (2015) show that cost-optimal planning is tractable for instances with bounded *domain size* and a polytree causal graph with bounded *diameter*, the length

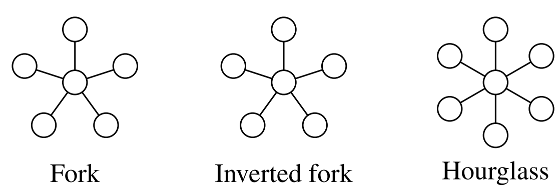

Figure 1: Some special cases of polytree graphs.

of the longest path in the underlying undirected graph. Clearly, all graphs in Figure 1 have diameter 2. 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 will 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).

In Sec. 2 we provide basic definitions for planning. In Sec. 3 we first derive a complexity bound for cost-optimal planning for polytree causal graphs based on the domain size and a bound B the number of variables changes in a plan. We then state such a bound as a function of the domain size and the depth of the causal graph, and combine this into the main theorem (Thm. 4). The actual bound is proven in Section 4. The paper concludes with a discussion section.

2 Preliminaries

We use the SAS$^+$ planning framework (Bäckström and Nebel 1995). Let V be a finite set of *variables* with a finite domain D of size $s = |D|$. The *state space* $S(V, D)$ is $D^{|V|}$ and the members of $S(V, D)$ are called *(total) states*. The *projection* $s[v]$ of a state s onto a variable v is the value of v in s. This can be viewed as a total function over V such that $s[v] \in D(v)$ for all $v \in V$. A *partial state* s is similarly a partial function over V such that for each $v \in V$, either $s[v]$ is undefined or $s[v] \in D(v)$. The notation vars(s) denotes the set of variables $v \in V$ such that $s[v]$ is defined. Projection is extended to sets of variables such that if $V' \subseteq V$, then $s[V']$ is a partial state that agrees with s on all variables in vars$(s) \cap V'$ and is otherwise undefined.

A *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 associated partial states the *precondition* pre(a) and the *effect* eff(a). Let $a \in A$ and let s be a total state. Then a is *valid in* s if pre$(a)[v] = s[v]$ for all $v \in$ vars(pre(a)). Furthermore, the *result of* a *in* s is a state $t \in S(V, D)$ such that for all $v \in V$, $t[v] =$ eff$(a)[v]$ if $v \in$ vars(eff(a)) and $t[v] = s[v]$ otherwise. Let $s_0, s_\ell \in S(V, D)$ and let $\omega = \langle a_1, \ldots, a_\ell\rangle$ be a sequence of actions. Then ω is a *plan from* s_0 *to* s_ℓ if either (1) $\omega = \langle\rangle$ and $\ell = 0$ or (2) there are states $s_1, \ldots, s_{\ell-1} \in S(V, D)$ such that for all i $(1 \leq i \leq \ell)$, a_i is valid in s_{i-1} and s_i is the result of a_i in s_{i-1}. An action sequence ω is a *plan for* $\mathbb{P}$ if it is a plan from s_I to some state s such that $s[v] = s_G[v]$ for all $v \in$ vars(s_G). The *length* of ω is $|\omega| = \ell$ and its *cost* is $c(\omega) = \sum_{i=1}^{\ell} c(a_i)$. We also define $C(v, \omega)$ for all $v \in V$ as the number of value changes of v when executing ω, i.e. $C(v, \omega)$ is the number of indices i $(1 \leq i \leq \ell)$ such that $s_i[v] \neq s_{i-1}[v]$. Let ω be a plan for $\mathbb{P}$. Then ω is a *shortest* plan for $\mathbb{P}$ if there is no plan ω' for $\mathbb{P}$ such that $|\omega'| < |\omega|$; ω is a *cost-optimal plan* for $\mathbb{P}$ if there is no plan ω' for $\mathbb{P}$ such that $c(\omega') < c(\omega)$; and ω is a *shortest cost-optimal plan* for $\mathbb{P}$ if it is cost optimal and there is no plan ω' 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.

We extend projections as follows. Let $V' \subseteq V$. For each $a \in A$, $a[V']$ is the restriction a' of a where pre(a') = pre$(a)[V']$ and eff(a') = eff$(a)[V']$. Also define $A[V'] = \{a[V'] \mid a \in A$ and vars(eff$(a[V'])) \neq \varnothing\}$ and $\mathbb{P}[V'] = \langle V', A[V'], s_I[V'], s_G[V']\rangle$. The projection of an action sequence $\omega = \langle a_1, \ldots, a_\ell\rangle$ over A onto V' is denoted $\omega[V']$ and defined as follows. First define the sequence $\omega' = \langle a'_1, \ldots, a'_\ell\rangle$ such that $a'_i = a_i[V']$ for all i $(1 \leq i \leq \ell)$. Then define $\omega[V']$ as the subsequence of ω' that contains only those a'_i where vars(eff$(a'_i)) \neq \varnothing$. For all cases, we also define projection onto a single variable v such that $a[v] = a[\{v\}]$ etc.

Each $v \in V$ has a *domain-transition graph* DTG$(v) = \langle D, T\rangle$, where for all $x, y \in D$, T contains an arc $\langle x, a, y\rangle$ if there is some $a \in A$ such that eff$(a)[v] = y$ and either pre$(a)[v] = x$ or $v \notin$ vars(pre(a)). 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 (1) $v \in$ vars(pre$(a)) \cup$ vars(eff(a)) and (2) $w \in$ vars(eff(a)) for some action $a \in A$. An action $a \in A$ is *unary* if $|$vars(eff$(a))| = 1$ and $\mathbb{P}$ is unary if all $a \in A$ are unary. It is immediate that $\mathbb{P}$ 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 ω 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, i.e. $d(v) = 0$ if v itself is a sink.

3 Planning for Polytrees

We will focus on planning for instances where the causal graph is a polytree. In this section, we will first derive a bound on the complexity of cost-optimal planning for such instances given that we know how many variable changes we must consider. We will then present such a bound as a function $B(s,d)$ of the domain size and the depth, satisfying that for every variable v and every shortest cost-optimal plan ω, 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 ω 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. First assume that all DTGs are acyclic. Let k be the maximum number of walks from $s_I[v]$ to $s_G[v]$ in DTG(v) over all $v \in V$. Bäckström (2014) shows how to decide if such an instance has a plan by making a tree decomposition of $CG(\mathbb{P})$ and then encode $\mathbb{P}$ as an instance $\mathbb{C}$ of the *Constraint Satisfaction Problem (CSP)*. Since the resulting primal graph of the CSP instance is a tree, this instance can be solved in time $O(N_{\mathbb{C}} D_{\mathbb{C}}^2)$ (Dechter and Pearl 1989), where $N_{\mathbb{C}}$ is the number of CSP variables and $D_{\mathbb{C}}$ is their domain size. When $CG(\mathbb{P})$ is a polytree, this whole process runs in time $O((ks)^6 n)$ (the dominating term in the proof of Corollary 10 in Bäckström (2014)). He also solves cost-optimal planning in this way by encoding $\mathbb{P}$ as an instance of the *Valued CSP (VCSP)* problem (Bäckström 2014, Theorem 12). No complexity figure is given here, but the only difference in the encoding is the additional weight function, so cost-optimal planning can also be solved in time $O((ks)^6 n)$ since VCSP with tree primal graphs can also be solved in time $O(N_{\mathbb{C}} D_{\mathbb{C}}^2)$ (Cooper and Schiex 2004, Theorem 5.4).

Bäckström remarks that the only reason for requiring acyclic DTGs is to bound the number of walks in the DTGs, and the CSP encoding does not require acyclic DTGs. In the encoding, each CSP variable x corresponds to a node in the tree decomposition, i.e. x corresponds to some subset $V' \subseteq V$, and the domain of x is all possible plans for the projection $\mathbb{P}[V']$. However, it is sufficient that the CSP variable domains are sets of subplans that guarantee there is a solution if $\mathbb{P}$ has a plan (possibly with certain properties). The parameter k is then the maximum number of walks we have to consider in any DTG under this condition. This observation was later exploited by Bäckström (2015).

Let B be a bound on $C(v,\omega)$ for all $v \in V$ over all shortest cost-optimal plans ω for $\mathbb{P}$. Let ω be such a plan. Since $CG(\mathbb{P})$ is a polytree, it follows from Prop. 1 that $|\omega[v]| = C(v,\omega) \leq B$ for all $v \in V$. Hence, we only have to consider walks up to length B and there are at most $\sum_{i=0}^{B}(s-1)^i \leq Bs^B$ such walks, so we can set $k = Bs^B$. Using the result above, we get that cost-optimal planning when $CG(\mathbb{P})$ is a polytree can be solved in time $O((ks)^6 n) \subseteq O((Bs^B s)^6 n) = O((Bs^{B+1})^6 n)$. $\square$

3.2 Bounds for Polytree Causal Graphs

In the derivation and definition of the bound function B, we need a function τ, 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

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

(we do not need the special cases where $s < 2$ or $h < 1$). For the special case where $s = 2$, we can simplify this expression to

$$\tau(2,h) = 1 + 2h,$$

and otherwise (when $s \geq 3$) we can rewrite it as

$$\begin{aligned}\tau(s,h) &= 1 + s\sum_{i=0}^{h-1}(s-1)^i = 1 + s\frac{(s-1)^h - 1}{(s-1)-1} \\ &= 1 + \frac{s}{s-2}(s-1)^h - \frac{s}{s-2}\end{aligned}$$

which can be upperbounded as

$$\tau(s,h) \leq 3(s-1)^h$$

since $1 \leq \frac{s}{s-2} \leq 3$ for all $s \geq 3$.

We can now define the bound function B of the domain size s and the depth d as

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

This function upper bounds the number of variable changes in all shortest cost-optimal plans.

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

The proof of the lemma appears in Sec. 4. We 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. $\square$

In order to get a better understanding of the function B, we will now study a non-recursive form of it. For $s \geq 3$, we can use the upper bound $\tau(s,h) \leq 3(s-1)^h$ to upper bound $B(s,d)$ as follows

$$\begin{aligned}B(s,d)+1 &= \tau\big(s, B(s,d-1)+1\big) + (s-2) + 1 \\ &\leq 3(s-1)^{B(s,d-1)+1} + (s-1) \\ &\leq 4(s-1)^{B(s,d-1)+1}.\end{aligned}$$

It follows that $B(s,d)$ can be upperbounded by a 'tower function' of the form

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

with d levels of exponentiation. We do not derive a corresponding non-recursive bound for the case where $s = 2$ since an even better bound will be derived at the end of Section 4.

4 Plan Lengths for Polytree Causal Graphs

This section is entirely devoted to the proof of Lemma 3. We first provide a higher-level overview of the proof to make it easier to understand the formal proof that follows.

4.1 Overview of the Proof

The proof assumes a planning instance $\mathbb{P} = \langle V, A, s_I, s_G, c\rangle$ with a polytree causal graph $CG(\mathbb{P})$. We will use the polytree in Figure 2 as an example, where the vertices are numbered $1, \ldots, 15$ for simplicity. Recall that the vertices of $CG(\mathbb{P})$ are the variables of $\mathbb{P}$ and we will interchangeably refer to them as vertices or variables. The graph is drawn such that all vertices with the same depth are aligned horizontally. The proof assumes an arbitrary shortest cost-optimal plan ω and shows by induction over the depth that all vertices satisfy the bound.

The base case is all vertices at depth 0, i.e. the sinks, which are vertices $1, \ldots, 5$ on the top row in the example. This case is straightforward since no other variables depend on the values of a sink, so the subplan for a sink only has to achieve the goal for this variable itself.

For the induction step, we must prove that if all vertices at depth d or less satisfy the bound, then also all vertices on depth $d+1$ satisfy the bound. Consider an arbitrary vertex u at depth $d+1$ and let $Out(u) = \{v_1, \ldots, v_m\}$ be the set of outgoing vertices of u. In the example, we assume the bound holds for all vertices at depth 0 and 1 and want to prove that it holds also for depth 2. We illustrate the case where we choose $u = 9$, which gives $Out(u) = \{2, 7\}$ and we choose $v_1 = 2$ and $v_2 = 7$ (the choice is arbitrary). Note that it follows from the definition of depth that at least one of $v_1, \ldots, v_m$ have depth $d = d(u) - 1$, but the other vertices may have a smaller depth. We see in the example that $d(v_2) = 2 = d(u) - 1$ but $d(v_1) = 0$.

Removing all outgoing arcs from u would split the graph into $m+1$ components $U, V_1, \ldots, V_m$ such that $u \in U$ and $v_i \in V_i$ for $1 \leq i \leq m$, i.e. these arcs are the only dependencies between the components. Note that $\{U, V_1, \ldots, V_m\}$ is a partition of V. In the example we get the sets $V_1 = \{1, 2, 6, 8, 11\}$, $V_2 = \{3, 4, 7\}$ and $U = \{5, 9, 10, 12, 13, 14, 15\}$, which clearly forms a partition of the vertex set. Note that this does not necessarily partition the vertices according to depth. For instance, V_1 contains vertex 11 with depth 3, which is larger than the depth of u, and U contains vertex 5 with depth 0, which is smaller than the depth of u. However, if a path from a source to a sink contains such a vertex, then it cannot also contain vertex u.

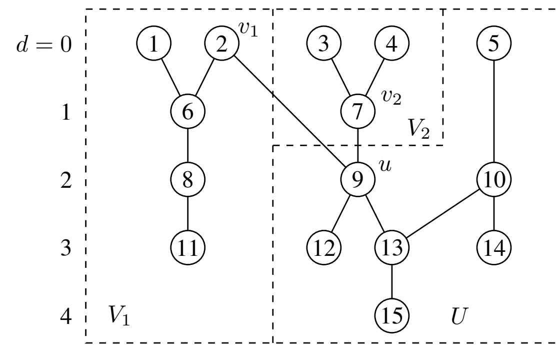

Figure 2: Example of partitioning a polytree into U, V_1, V_2, for a vertex u with two out vertices v_1 and v_2.

Since there are no arcs between the sets $V_1, \ldots, V_m$, these are all independent of each other. Furthermore, the only dependencies between U and $V_1, \ldots, V_m$ are the arcs from u to $v_1, \ldots, v_m$. Because of these limited dependencies, we can prove the bound independently for each vertex at depth $d+1$, so the proof only needs to consider an arbitrary such vertex u. The actual proof of the bound for u then proceeds in three steps, I–III.

In step I we exploit the limited dependencies between the parts $U, V_1, \ldots, V_m$ and split the plan ω into the corresponding subplans $\omega[U], \omega[V_1], \ldots, \omega[V_m]$. Since all actions are unary, these subplans have no actions in common so they form a kind of partition of ω. Furthermore, the restricted dependencies makes it possible to split ω into these subplans and then reassemble them in a different order as long as the dependencies between $\omega[U]$ and each one of $\omega[V_1], \ldots, \omega[V_m]$ are satisfied. In particular, we show that there exists such a reordered plan ω' where all actions in $\omega[V_1], \ldots, \omega[V_m]$ that depend on some variable in U occur as early as possible in the new plan. Since ω and ω' have exactly the same actions, they have the same length and cost, but the properties of ω' makes it easier to analyse than ω.

In step II we exploit that only the actions in the subplans $\omega[v_1], \ldots, \omega[v_m]$ can depend on variables in U and that u is the only such variable they can depend on. Let $\chi_1, \ldots, \chi_m$ be the corresponding sequences of precondition values on u in the subplans $\omega[v_1], \ldots, \omega[v_m]$ and let ψ be the sequence of values that u has during the plan. Since all actions must have satisfied preconditions, it follows that all of $\chi_1, \ldots, \chi_m$ are subsequences of ψ. Figure 3 illustrates an example with four sequences $\chi_1, \ldots, \chi_4$ and how these can be mapped to subsequences of ψ. We then define a tree T that contains all prefixes of the sequences $\chi_1, \ldots, \chi_m$ and all nodes of this tree must then occur as subsequences of ψ. The prefix tree for the four sequences in Figure 3 is drawn in black.

In step III we prove the actual bound on the number of variable changes, by proving a bound on the length of ψ. We do this by embedding the prefix tree T into another prefix tree $T^{s,h+1}_{\text{max}}$ of known size. We then show that the prefixes of T can be mapped to ψ in such a way that for every element of ψ that does not contribute to satisfying some new prefix in T, the size of T is correspondingly smaller than $T^{s,h+1}_{\max}$. This analysis is then used to show that the length of ψ is upper bounded by the size of $T^{s,h+1}_{\max}$. It is shown in Figure 3 how T is embedded as a subtree of $T^{s,h+1}_{\max}$, where the additional arrows and nodes of the latter are drawn in grey.

4.2 Proof

Proof of Lemma 3. Let $\mathbb{P} = \langle V, A, s_I, s_G, c\rangle$ be an arbitrary solvable SAS$^+$ instance with a polytree causal graph. Let ω be an arbitrary shortest cost-optimal plan for $\mathbb{P}$. Proof by induction over the depth d, that for all $d \geq 0$ it holds that $C(v, \omega) \leq B(s, d(v))$ for all $v \in V$ such that $d(v) \leq d$.

Base case: We have $d = 0$ in the base case. Let $v \in V$ be an arbitrary variable with $d(v) = 0$ in $CG(\mathbb{P})$. Then v is a sink so $Out(v) = \varnothing$ and, thus, no other variable depends on it. Hence, the plan $\omega[v]$ only needs to be a walk in DTG(v) from $s_I[v]$ to $s_G[v]$. A cycle in $\omega[v]$ cannot make ω shorter or cheaper, so in the worst case, $s_I[v]$ is a Hamilton path, i.e. of length $s - 1$. It follows that $C(v, \omega) \leq s - 1 = B(s, 0)$ for all $v \in V$ with $d(v) = 0$, which proves the base case.

Induction: Suppose the claim holds for some $d \geq 0$, i.e. $C(v, \omega) \leq B(s, d(v))$ for all $v \in V$ with $d(v) \leq d$. We must prove that $C(v, \omega) \leq B(s, d(v))$ also for all $v \in V$ with $d(v) = d + 1$. Let $u \in V$ be an arbitrary variable such that $d(u) = d+1$ and assume that $Out(u) = \{v_1, \ldots, v_m\}$. Let $U, V_1, \ldots, V_m$ be the corresponding partition of V as explained above. To enhance readability, we will define names for the subplans of ω defined by the projections of ω onto these sets as follows:

$$\begin{aligned} \mu &= \omega[U] \\ \omega_i &= \omega[V_i] \text{ for } 1 \leq i \leq m \end{aligned}$$

That is, μ contains the subsequence of all actions that have an effect on some variable in U (since there is a one-to-one correspondence between the vertices and the variables of $\mathbb{P}$), and each ω_i is analogously the subsequence of actions with an effect on some variable in V_i. Recall that since a polytree is acyclic, all actions must be unary, i.e. each action can have an effect on one variable only. Since $U, V_1, \ldots, V_m$ is a partition of the variable set V, it follows that the subsequences $\mu, \omega_1, \ldots, \omega_m$ partition ω in the sense that every action in ω occurs in exactly one of the subsequences $\mu, \omega_1, \ldots, \omega_m$.

We will be particularly interested in the subsequences corresponding to variables $u, v_1, \ldots, v_m$. First note that we can write μ on the form

$$\mu = \omega[U] = \mu^0, b^1, \mu^1, b^2, \mu^2, \ldots, b^\ell, \mu^\ell,$$

where the subsequence

$$\beta = \mu[u] = b^1, \ldots, b^\ell$$

are the actions that have a defined effect on u. That is, the actions in β have an effect on u only and the actions in the sequences $\mu^0, \ldots, \mu^\ell$ have effects on the variables in $U \setminus \{u\}$ only. For each i, we can similarly write ω_i on the form

$$\omega_i = \omega[V_i] = \omega_i^0, a_i^1, \omega_i^1, a_i^2, \omega_i^2, \ldots, a_i^{h_i}, \omega_i^{h_i},$$

where the subsequence

$$\alpha_i = \omega_i[v_i] = a_i^1, a_i^2, \ldots, a_i^{h_i},$$

are the actions in ω_i that have a defined effect on v_i and where h_i is the length of α_i. That is, the actions in α_i have an effect on v_i only and the actions in the sequences $\omega_i^0, \ldots, \omega_i^{h_i}$ only have effect on the variables in $V_i \setminus \{v_i\}$.

We also define names for the following sequences of domain values. For each i ($1 \leq i \leq m$), let

$$\chi_i = x_i^1, \ldots, x_i^{h_i} = \text{pre}(a_i^1)[u], \ldots, \text{pre}(a_i^{h_i})[u],$$

i.e. the sequence of precondition values that the actions in α_i have on u. Also let

$$\psi = y^0, \ldots, y^\ell,$$

corresponding to β such that $y^0 = s_I[u]$ and $y^k = \text{eff}(b^k)[u]$ for $1 \leq k \leq \ell$, i.e. ψ is the sequence of values of variable u while executing ω. For each i ($1 \leq i \leq m$), every defined values in χ_i must be achieved by the initial state or some action in β. Consider two consecutive values x_i^t, x_i^{t+1} in χ_i. If both are defined and have different values, then β must contain at least one action between a_i^t and a_i^{t+1} to change u, while this is not necessary if $x_i^t = x_i^{t+1}$ or if either or both are undefined. We will later want to prove an upper bound on the length of ψ in the worst case, so we can assume for all i ($1 \leq i \leq m$) that

1. x_i^t is defined for all t ($1 \leq t \leq h_i$) and
2. $x_i^t \neq x_i^{t+1}$ for all t ($1 \leq t < h_i$).

Under this assumption, it obviously also holds that each of the sequences $\chi_1, \ldots, \chi_m$ appear as subsequences of ψ. We also define $h = \max_i h_i$ and $S = \{\chi_i \mid 1 \leq i \leq m\}$. Given a plan ω, we will also use the shorthand $a_i \prec_\omega a_j$ when $i < j$[1].

The remainder of the proof will be in three steps:

I) We define a 'release-time function' r for ω that maps $\chi_1, \ldots, \chi_m$ to ψ. Then we show that ω can be reordered into an equivalent plan ω' with such a function r' with certain minimality properties.

II) We define a tree T of all prefixes of $\chi_1, \ldots, \chi_m$ and a 'tree mapping' r_T from T to ψ, and show that we can use function r' above to construct such a mapping r'_T with certain minimality properties. We also define the maximal prefix tree $T^{s,h}_{\max}$ of height h.

III) We show that the length of ψ is bounded by the size of $T^{s,h+1}_{\max}$ by embedding T as a subtree of $T^{s,h+1}_{\max}$ and compare their sizes. This also yields the bound for $C(u, \omega)$.

Step I. We know that:

[1]Formally, this definition requires that a plan is defined as a sequence of action instances, rather than actions. In order to simplify the presentation we do not explicitly make this distinction, trusting the reader to interpret correctly. Furthermore, since we will only consider a given plan and reorderings of it, we can alternatively assume that all actions in a plan are unique, even if they may have the same definitions.

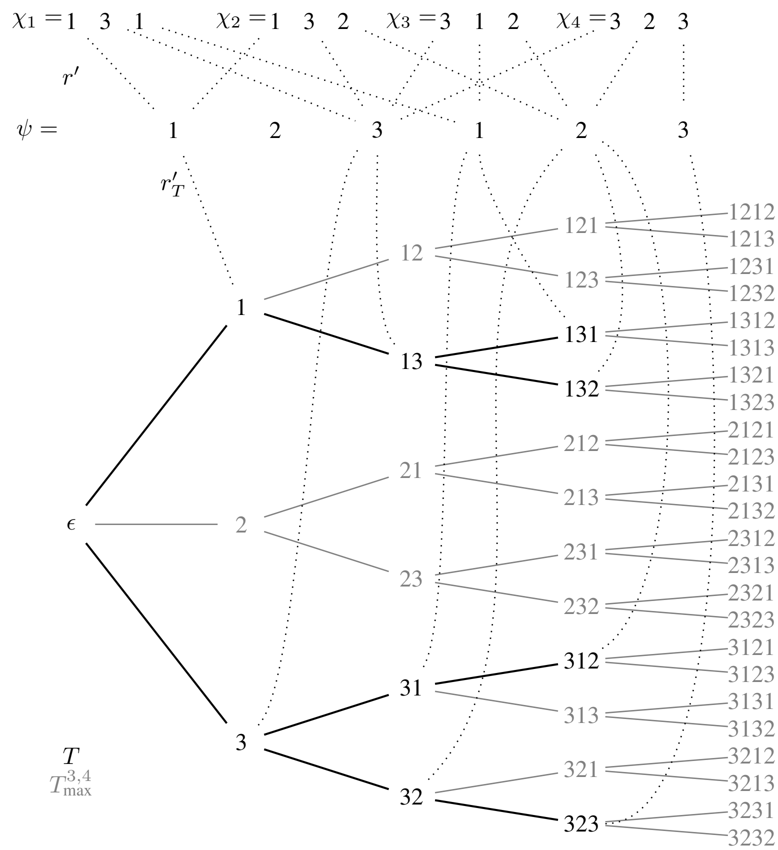

Figure 3: Example illustrating the induction part of the proof of Lemma 3. We have $S = \{\chi_1, \ldots, \chi_4\}$, where $\chi_1 = 131$, $\chi_2 = 132$, $\chi_3 = 312$ and $\chi_4 = 323$, and we have $\psi = 123123$. The maximal prefix tree $T^{3,4}_{\text{max}}$ is drawn in grey with the prefix tree T for S drawn on top of it with solid black arcs to illustrate that T is a subtree of $T^{3,4}_{\text{max}}$. The example also shows the minimal release-time function r' from S to ψ and a minimal tree mapping r'_T from T to ψ with dotted arrows.

1. All actions are unary, i.e. have effect on one variable only.
2. All actions in μ have preconditions on variables in U only.
3. For each i, the only actions in ω_i that can have a precondition on some variable outside V_i are the actions in α_i, which can have a precondition also on variable u.

It follows from 1 and 3 that every interleaving of $\omega_1, \ldots, \omega_m$ must be a plan for $\mathbb{P}[V_1 \cup \ldots \cup V_m] = \mathbb{P}[V \setminus U]$. Combining this with 2 yields that an interleaving of $\mu, \omega_1, \ldots, \omega_m$ is a plan for $\mathbb{P}$ if all actions in $\alpha_1, \ldots, \alpha_m$ have their preconditions on u satisfied. In particular, we will show below that we can make such an interleaving which satisfies a certain minimality property.

The condition that all actions actions in $\alpha_1, \ldots, \alpha_m$ have their preconditions on u satisfied in ω can be modelled by a *'release-time function'* r that maps all pairs i, t ($1 \leq i \leq m$, $1 \leq t \leq h_i$) to $\{0, \ldots, \ell\}$, such that:

1. $r(i, 1) = 0$ if $a^1_i \prec_\omega b^1$,
2. $r(i, h_i) = \ell$ if $b^\ell \prec_\omega a^{h_i}_i$ and otherwise
3. $r(i, t) = k$ where $b^k \prec_\omega a^t_i \prec_\omega b^{k+1}$ for $1 \leq t \leq h$.

One may think of $r(i, t)$ as the 'release time' of action a^t_i in ω with respect to the actions in β, i.e. $r(i, t)$ does not specify an exact position in ω, but only a sufficient ordering between a^t_i and β. Case 1 only occurs when $x^1_i = y^0$. Otherwise, b^k sets the precondition on u for a^t_i in ω. Note that $x^1_i = y^0$ may hold also in this case, if $y^{r(i,1)} = y^0$; the actual value depends on ω. It must hold that $y^{r(i,t)} = x^t_i$ for all i and t so r is also a mapping from the sequences in S to ψ, constituting a witness that all $\chi_i \in S$ are subsequences of ψ.

We will furthermore say that a release-time function is minimal if all actions in $\alpha_1, \ldots, \alpha_m$ occur as early as possible. Formally, r is a *minimal release-time function* if it also satisfies that for all i ($1 \leq i \leq m$):

1. $r(i, 1) = k$, for the minimum k such that $y^k = x^1_i$, and
2. $r(i, t) = k$, for the minimum $k > r(i, t-1)$ such that $y^k = x^t_i$, for $1 < t \leq h_i$.

The release-time function r' in in Figure 3 is minimal. Note that minimality is defined with respect to S and ψ, not a particular plan, so there is always a unique minimal release-time function, but not every plan satisfies it.

Using the previous observation on interleaving of plans, we will now show that we can always reorder the subplans $\mu, \omega_1, \ldots, \omega_m$ with respect to each other such that the result is a plan with a minimal release-time function.

Claim 5. *There exists a plan ω' for $\mathbb{P}$ with release-time function r' such that ω' is a reordering of ω and r' is minimal.*

Proof of the claim. First define r' as a function satisfying the minimality criteria above. We must show that we can construct ω' as an interleaving of $\mu, \omega_1, \ldots, \omega_m$ such that r' is the release-time function for ω'. This exploits the observation above that $\omega_1, \ldots, \omega_m$ are independent of each other, so it is only necessary to order each of them with respect to μ. Furthermore, they only need to be ordered with respect to β, since $\mu[U \setminus \{u\}]$ does not interact with $\omega_1, \ldots, \omega_m$.

First interleave $\alpha_1, \ldots, \alpha_m$ with β, using r' as a guide. Start with $\omega' = \beta$. For each i and t ($1 \leq i \leq m, 1 \leq t \leq h_i$), insert a_i^t at any position between $b^{r'(i,t)}$ and $b^{r'(i,t)+1}$, alternatively before b^1 if $r'(i,1) = 0$ or after b^ℓ if $r'(i,h_i) = \ell$. This preserves the internal orders of $\alpha_1, \ldots, \alpha_m$ and β and each a_i^t has its precondition on u satisfied, so this is a plan for $\mathbb{P}[\{u, v_1, \ldots, v_m\}]$. Then for each i ($1 \leq i \leq m$), interleave the remainder of $\omega_1, \ldots, \omega_m$ with ω' as follows. Insert ω_i^0 anywhere before a_i^1, insert $\omega_i^{h_i}$ anywhere after $a_i^{h_i}$ and for each t ($1 \leq t < h_i$), insert ω_i^t anywhere between a_i^t and a_i^{t+1}. None of these subplans have any variables in common with β. Furthermore, since $\omega_i, \ldots, \omega_m$ have only variable u in common and can only have preconditions on it, they cannot interfere with each other. Since the internal orders of $\omega_1, \ldots, \omega_m$ are preserved, this is a plan for $\mathbb{P}[\{u\} \cup V_1 \cup \ldots \cup V_m]$.

Finally, we interleave the remainder of μ with ω' as follows. Insert μ^0 anywhere before b^1, insert μ^ℓ anywhere after b^ℓ and for each k ($1 \leq k < \ell$), insert μ^k anywhere between b^k and b^{k+1}. This preserves the internal order of μ and since none of $\mu^0, \ldots, \mu^\ell$ share any variables with $\omega_1, \ldots, \omega_m$, it follows that ω' is a plan for $\mathbb{P}$. $\square$

Since ω' is a reordering of ω it has exactly the same actions with the same number of occurences, so the two plans have the same length and cost, i.e. either both ω and ω' are shortest cost-optimal plans or none of them is. In particular, it holds that $\omega'[v] = \omega[v]$ for all $v \in V$, so the sequences $\chi_1, \ldots, \chi_m$ and ψ are the same for both plans. Hence, we can analyse ω' instead of ω. Note that this reordering is possible due to the inherent restrictions of polytrees.

Step II. We will now define the concept of prefix trees and tree mappings. Given a sequence σ and a value t ($0 \leq t \leq |\sigma|$), we define the t-prefix of σ as the prefix consisting of the t first elements in σ. For every $\chi_i \in S$ and every t ($1 \leq t \leq h_i$), let ρ_i^t be the t-prefix of χ_i, and let P be the set of all such prefixes, including the empty prefix ϵ. Then define the *prefix tree* T for P as a directed tree $T = \langle N, F \rangle$ as follows:

1. The node set N contains a node n_ρ for each prefix $\rho \in P$.
2. The arc set F contains an arc $\langle n_\rho, n_{\rho'} \rangle$ for each pair of prefixes $\rho, \rho' \in P$ such that $\rho' = \rho x$ for some $x \in D$.

It follows that n_ϵ is the root of T. One may think of a node n_ρ as representing the set of all $\chi_i \in S$ that have the same t-prefix ρ, where $t = |\rho|$. All values in D are possible in the first position of the prefixes and otherwise it follows from the worst-case assumption above that the values must alternate. Hence, the root has at most s children and all other nodes have at most $s - 1$ children. The height of T is the length of the longest prefix, i.e. h. An example of a prefix tree T appears in Figure 3.

Also let $T_{\max}^{s,h}$ denote the largest possible prefix tree of height h for domain size s, which is the prefix tree for the set of all possible alternating sequences of length h over the domain. The root in this tree has exactly s children and every other interior node has exactly $s - 1$ children, so it follows that $T_{\max}^{s,h}$ has exactly $\tau(s, h)$ nodes. Also note that any prefix tree of height h, or less, must be a subtree of $T_{\max}^{s,h}$.

A prefix tree T represents a strict partial order $\prec_T$ on N such that $n_\rho \prec_T n_{\rho'}$ if n_ρ is an ancestor of $n_{\rho'}$, i.e. if ρ is a proper prefix of ρ'. Define a function $v : N \to D$ such that $v(n_\epsilon) = y^0$ and otherwise $v(n_\rho)$ is the value in the last position of ρ. A function $r_T : N \to \{0, \ldots, \ell\}$ is a *tree mapping* of T to ψ if it holds that:

1. $y^{r_T(n)} = v(n)$ for all $n \in N$,
2. if $n \prec_T n'$ then $r_T(n) < r_T(n')$, for all $n, n' \in N$.

Note that nodes along the same branch of T must be ordered in strictly increasing order by r_T, while it is possible that nodes on different branches are mapped to the same index. Furthermore, r_T is a *minimal tree mapping* if

1. $r_T(n_\epsilon) = 0$ and
2. for every $\langle n, n' \rangle \in F$, $r_T(n')$ is the smallest index k such that $r_T(n) < k$ and $y^k = v(n')$.

The tree mapping r'_T in the example in Figure 3 is minimal.

It is immediate from the definition of tree mappings that if a tree mapping exists from T to ψ, then every prefix $\rho \in P$ is a subsequence of ψ. We will now show that also the opposite holds, but we first need the following result.

Claim 6. *If r is a minimal release-time function, then it has the following property: for all $\chi_i, \chi_j \in S$ and all t ($1 \leq t \leq \min\{h_i, h_j\}$), if χ_i and χ_j have the same t-prefix, then $r(i,t) = r(j,t)$.*

Proof of the claim. Proof by induction over t.

Base case: For $t = 1$, we have $x_i^1 = x_j^1$ so it is immediate from the minimality definition that $r(i,1) = r(j,1)$.

Induction: Suppose the claim holds for some t. Let χ_i and χ_j have the same (t+1)-prefix. Then $x_i^t = x_j^t$ so it follows from the induction hypothesis that $r(i,t) = r(j,t)$. It also holds that $x_i^{t+1} = x_j^{t+1}$ so it follows from the minimality definition that also $r(i,t+1) = r(j,t+1)$. $\square$

We can now prove that a minimal tree mapping must exist.

Claim 7. *If all $\rho \in P$ are subsequences of ψ, then there exists a minimal tree mapping r'_T from T to ψ.*

Proof of the claim. Let r' be the minimal release-time function. We define a function $r'_T : N \to \{0, \ldots, \ell\}$ based on r' as follows. Let $r'_T(n_\epsilon) = 0$. For every other node $n_\rho \in N$, let $t = |\rho|$ and let $r'_T(n_\rho) = r'(i, t)$ for some $\chi_i \in S$ that has ρ as a t-prefix. The choice does not matter since r' assigns the same value for all such χ_i (Claim 6). We must show that r_T is a tree mapping from T to ψ.

For condition 1, let $n_\rho \in N \setminus \{n_\epsilon\}$ be an arbitrary node and let $t = |\rho|$. Then ρ must be a t-prefix of some $\chi_i \in S$, so $v(n_\rho) = x_i^t$. We get that $y^{R_T(n_\rho)} = y^{r'(i,t)} = x_i^t = v(n_\rho)$. The condition is trivially satisfied for n_ϵ.

For condition 2, suppose $n_\rho \prec_T n_{\rho'}$. Then ρ is a proper prefix of ρ'. Let $t = |\rho|$ and $t' = |\rho'|$. Let $\chi_i \in S$ be any sequence having ρ' as a prefix. Then $r'_T(n_\rho) = r'(i, t)$ and $r'_T(n_{\rho'}) = r'(i, t')$, so it follows that $r'_T(n_\rho) < r'_T(n_{\rho'})$ since $r'(i, t) < r'(i, t')$ must hold.

Finally, suppose that r'_T is not minimal. Then there must exist some $\langle n_\rho, n_{\rho'}\rangle \in F$ and an index k such that $r'_T(n_\rho) < k < r'_T(n_{\rho'})$ and $y^k = v(n_{\rho'})$. Let $\chi_i \in S$ be any sequence with ρ' as a t-prefix. Then χ_i also has ρ as a $(t-1)$-prefix, so $r'(i, t-1) = r'_T(n_\rho)$ and $r'(i, t) = r'_T(n_{\rho'})$. However, then $y^k = x_i^t$, which contradicts that r' is minimal. It follows that also r'_T must be minimal. □

Step III. We are now ready to prove a bound on the length of ψ, which implies the bound on $C(u, \omega)$.

Claim 8. $|\psi| \leq \tau(s, h+1) + (s-1)$.

Proof of the claim. Let r'_T be the minimal tree mapping in the proof of Claim 7. Let $R = \{r'_T(n) \mid n \in N\}$ and let $r_1, \ldots, r_{|R|}$ be the values in R in increasing order. That is, R is the set of all indices in ψ that r'_T maps some node to and $y^{r_{|R|}}$ is the last position in ψ that is used to satisfy some precondition in $\alpha_1, \ldots, \alpha_m$. Let $\psi' = y^0 \ldots y^{r_{|R|}}$, i.e. ψ' is the shortest prefix of ψ satisfying S. Let $\Delta = |\psi'| - |R|$, i.e. the number of elements in ψ' that r'_T does not map any node to.

Recall that T is a subtree of the maximal prefix tree $T^{s,h}_{\max}$, and thus also of $T^{s,h+1}_{\max}$. We will determine an upper bound on the length of ψ' based on comparing the sizes of T and $T^{s,h+1}_{\max}$. For every consecutive pair $r_i, r_{i+1} \in R$, arbitrarily choose nodes $n_i, n_{i+1} \in N$ such that $R'_T(n_i) = r_i$ and $R'_T(n_{i+1}) = r_{i+1}$ (note that there can be more than $|R|$ nodes in T, but r'_T does not map these to more than $|R|$ different indices). Then there can be no $n \in N$ such that $r_i < R'_T(n) < r_{i+1}$. Hence, none of the elements $y^{r_i}, \ldots, y^{r_{i+1}}$ is used to satisfy S, i.e. to satisfy any precondition of the actions in $\alpha_1, \ldots, \alpha_m$. It follows that a cycle in $y^{r_i}, \ldots, y^{r_{i+1}}$ would be redundant, contradicting that ω' is a shortest cost-optimal plan, so no value occurs more than once in this sequence. There are thus $r_\delta = r_{i+1} - r_i - 1$ values between positions r_i and r_{i+1} and they contribute the amount r_δ to Δ. None of these values can be the value of a child of n_i since R'_T is minimal, so node n_i has r_δ unique subtrees in $T^{s,h+1}_{\max}$ that n_i does not have in T. Note that this holds also if n_i is a leaf in T since $T^{s,h+1}_{\max}$ has height $h+1$ and n_i thus cannot be a leaf in $T^{s,h+1}_{\max}$. The size of each such subtree is at least 1, so together they reduce the size of T with at least r_δ compared to the size of $T^{s,h+1}_{\max}$. We know that $T^{s,h+1}_{\max}$ has $\tau(s, h+1)$ nodes, so it follows that $|N| \leq \tau(s, h+1) - \Delta$. Since $|R| \leq |N|$ we get $|\psi'| = |R| + \Delta \leq (\tau(s, h+1) - \Delta) + \Delta = \tau(s, h+1)$.

The remainder $y^{r_{|R|}+1}, \ldots, y^\ell$ of ψ serves only to satisfy the goal. If this suffix is longer than $s-1$, then it must contain a redundant cycle which contradicts that ω', and thus ω, is a shortest cost-optimal plan. It follows that $|\psi| \leq \tau(s, h+1) + (s-1)$. □

This proof is illustrated in Fig. 3, where r'_T does not map any node in T to the first occurence of 2 in ψ Hence, the whole subtree rooted at node 2 in $T^{s,h+1}_{\max}$ is missing in T. Note that also some more subtrees are missing, without contributing to the length of ψ, i.e. these reduce the length of ψ further than the analysis in the proof above shows.

It remains to prove that $C(u, \omega) \leq B(s, d(u))$. We know that $C(u, \omega) = |\beta| = |\psi| - 1$, from Prop. 1, and that $C(v_i, \omega) = h_i$ for all i ($1 \leq i \leq m$). We also know from the induction hypothesis that $C(v_i, \omega) \leq B(s, d(v_i)) \leq B(s, d)$ so we get that $h \leq B(s, d)$ and, thus, that

$$\begin{aligned} C(u, \omega) &= |\psi| - 1 \\ &\leq \tau(s, h+1) + (s-1) - 1 \\ &= \tau(s, h+1) + s - 2 \\ &\leq \tau(s, B(s, d) + 1) + s - 2 \\ &= B(s, d+1). \end{aligned}$$

This ends the induction and, thus, the proof of Lemma 3. □

For the special case where $s = 2$, we can easily get a better bound. Assume $D = \{1, 2\}$. Then all of $\chi_1, \ldots, \chi_m$ and ψ must be alternating sequences of 1 and 2, starting with either 1 or 2. Every $\chi_i \in S$ is then equivalent to, or a prefix of, either $y^0, \ldots, y^{h-1}$ or $y^1, \ldots, y^h$. One final alternation may be necessary to satisfy the goal, so we get that $|\psi| \leq h+2$ in this case, i.e. $\beta \leq h+1$. For this special case, we can thus improve the bound to $B(2, d) = d+1$.

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.

Another possible generalisation is to relax the domain size bound to hold only for certain variables, which Katz and Keyder (2012) used to prove that planning for fork causal graphs is tractable when only the root variable is bounded.

Although our algorithm is polynomial-time, its running time is admittedly not impressive. However, previous tractability results like Aghighi, Jonsson, and Ståhlberg (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. Otherwise, to construct substantially faster algorithms may require that they are based on some other principle than the VCSP transformations used in the literature, cf. Bäckström (2014), Brafman and Domshlak (2006), and Cooper, de Roquemaurel, and Régnier (2011).

Acknowledgments

We are grateful to the reviewers for very detailed and valuable comments. Bäckström is partially supported by the Swedish Research Council (VR) under grant 621-2014-4086.

References

Aghighi, M.; Jonsson, P.; and Ståhlberg, S. 2015. Tractable cost-optimal planning over restricted polytree causal graphs. In *Proc. 29th AAAI Conference on Artificial Intelligence (AAAI 2015), Austin, TX, USA*, 3225–3231.

Bäckström, C., and Nebel, B. 1995. Complexity results for SAS^+ planning. *Comput. Intell.* 11:625–656.

Bäckström, C. 2014. 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.*, 33–38.

Bäckström, C. 2015. Some fixed parameter tractability results for planning with non-acyclic domain-transition graphs. In *Proc.29th AAAI Conf. Artif. Intell., 2015, Austin, TX, USA.*, 3232–3238.

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

Brafman, R. I., and Domshlak, C. 2013. On the complexity of planning for agent teams and its implications for single agent planning. *Artif. Intell.* 198:52–71.

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

Cooper, M. C.; de Roquemaurel, M.; and Régnier, P. 2011. A weighted CSP approach to cost-optimal planning. *AI Commun.* 24(1):1–29.

Cooper, M.; Maris, F.; and Régnier, P. 2014. Monotone temporal planning: Tractability, extensions and applications. *J. Artif. Intell. Res.* 50:447–485.

Dechter, R., and Pearl, J. 1989. Tree clustering for constraint networks. *Artif. Intell.* 38(3):353–366.

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

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

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

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

Helmert, M.; Haslum, P.; and Hoffmann, J. 2007. Flexible abstraction heuristics for optimal sequential planning. In *Proc.17th Int'l Conf. Automated Planning and Scheduling, ICAPS 2007, Providence, RI, USA, 2007*, 176–183.

Helmert, M. 2004. A planning heuristic based on causal graph analysis. In *ICAPS*, 161–170.

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

Jonsson, A.; Jonsson, P.; and Lööw, T. 2014. Limitations of acyclic causal graphs for planning. *Artif. Intell.* 210:36–55.

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

Katz, M., and Domshlak, C. 2010. Implicit abstraction heuristics. *J. Artif. Intell. Res.* 39:51–126.

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

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

Muscettola, N.; Nayak, P. P.; Pell, B.; and Williams, B. C. 1998. Remote agent: To boldly go where no AI system has gone before. *Artif. Intell.* 103(1-2):5–47.

Ståhlberg, S. 2017. *Methods for Detecting Unsolvable Planning Instances using Variable Projection*. Ph.D. Dissertation, Linköping University, Sweden.

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