Structural Patterns Beyond Forks: Extending the Complexity Boundaries of Classical Planning

Michael Katz
Saarland University
Saarbrücken, Germany
katz@cs.uni-saarland.de

Emil Keyder
INRIA
Nancy, France
emilkeyder@gmail.com

Abstract

Tractability analysis in terms of the causal graphs of planning problems has emerged as an important area of research in recent years, leading to new methods for the derivation of domain-independent heuristics (Katz and Domshlak 2010). Here we continue this work, extending our knowledge of the frontier between tractable and NP-complete fragments. We close some gaps left in previous work, and introduce novel causal graph fragments that we call the *hourglass* and *semifork*, for which under certain additional assumptions optimal planning is in P. We show that relaxing any one of the restrictions required for this tractability leads to NP-complete problems. Our results are of both theoretical and practical interest, as these fragments can be used in existing frameworks to derive new abstraction heuristics. Before they can be used, however, a number of practical issues must be addressed. We discuss these issues and propose some solutions.

Introduction

Quantifying the complexity of classical planning problems in terms of their structure has long been an important research problem. Recent work in this area has focused on *causal graphs* (Domshlak and Dinitz 2001; Brafman and Domshlak 2003; Chen and Giménez 2008; Katz and Domshlak 2008; 2010; Giménez and Jonsson 2008), directed graphs whose nodes represent the variables of the problem and whose edges give information about dependencies between variables (Knoblock 1994). Combining limitations on causal graph structure with further restrictions on the sizes of variable domains and k-dependence, defined as the maximum number of variables on which an action has preconditions while not changing their values, has led to complexity results that apply to a wide range of problems (Katz and Domshlak 2008; Giménez and Jonsson 2009). Such results are not of purely theoretical interest, as the causal graph is used in a variety of practical applications from problem decomposition (Brafman and Domshlak 2006) to the derivation of non-admissible domain-independent heuristics for satisficing planning (Helmert 2004).

The work we present here is motivated by a different use of tractable fragments of the causal graph: the derivation of *admissible* domain-independent heuristics. Search with such heuristics is one of the most successful approaches to optimal planning, and an important advance in this field over the last few years has been the introduction of *structural pattern heuristics* (Katz and Domshlak 2010). The idea behind these heuristics is to project planning problems onto fragments of causal graphs known to be tractable for optimal planning, and to use the costs of solutions to these as guidance for the original problem. Structural pattern heuristics play an important theoretical role in optimal planning, as they represent one of the handful of existing ideas for deriving admissible heuristics (Helmert and Domshlak 2009).

The usefulness of structural pattern heuristics increases directly with the availability of causal graph fragments that are known to be solvable optimally in polynomial time. Until now, they have made use of two non-trivial structures known as the *fork* and the *inverted fork*. Our principal aim here is to discover the limits of tractability for these two structures, removing restrictions and considering wider classes of causal graphs until the point at which optimal planning becomes NP-complete is found. This approach allows us to close several gaps in previous work, and results in the introduction of two new classes that under certain limitations are tractable for optimal planning and can be used in such heuristics, *hourglasses* and *semiforks*. We also show that the relaxation of any one of the assumptions required for this tractability leads to an NP-complete problem. While the use of these classes in structural pattern heuristics could improve their estimates, a number of practical issues remain to be solved before they can be adapted to that context. We briefly discuss these issues, and propose some solutions.

Preliminaries

We consider planning problems in the $\textsc{sas}^+$ formalism (Bäckström and Nebel 1995), given by a quintuple $\Pi = \langle V, A, I, G, cost \rangle$ where:

- V is a set of *state variables*, each $v \in V$ associated with a finite domain $\mathcal{D}(v)$. The value assigned to a variable v by a (possibly partial) assignment p to V is denoted by $p[v]$. A complete assignment s to V is called a *state*, and the set of all possible complete assignments S is the *state space* of Π. I is the *initial state*. The *goal* G is a partial assignment to V; a state s is a *goal state* iff $G \subseteq s$.
- A is a finite set of *actions*, each action $a \in A$ given by

a pair $\langle \mathsf{pre}(a), \mathsf{eff}(a) \rangle$ of partial assignments to V called *preconditions* and *effects*, respectively. By $A_v \subseteq A$, we denote the actions changing the value of v. $cost : A \rightarrow \mathbb{R}^{0+}$ is a real-valued, non-negative *cost* function.

An action a is applicable in a state s iff $\mathsf{pre}(a) \subseteq s$. The state s' resulting from applying a in s is denoted by $s[\![a]\!]$ and differs from s in that $s[v] = \mathsf{eff}(a)[v]$ whenever this is defined. $s[\![\langle a_1, \ldots, a_k \rangle]\!]$ denotes the state resulting from sequential application of the actions $a_1, \ldots, a_k$ in s. Such an action sequence is an *s-plan* if $G \subseteq s[\![\langle a_1, \ldots, a_k \rangle]\!]$, and it is an *optimal* s-plan if the summed cost $\sum_{i=1}^{k} cost(a_i)$ is minimal among all s-plans. The aim of (optimal) planning is to find an (optimal) I-plan. In what follows, we denote a plan for state s with $\pi(s)$ or just π when s is clear from the context, and use the notation π^* to specify that a plan is optimal. h^* denotes the cost of such an optimal plan.

The *causal graph* of Π is a digraph $\mathit{CG}(\Pi) = \langle V, E \rangle$ over the set of nodes V that contains an arc (v, v') iff $v \neq v'$ and there exists $a \in A$ such that $\mathsf{eff}(a)[v']$ and either $\mathsf{pre}(a)[v]$ or $\mathsf{eff}(a)[v]$ is specified. Given a variable v, we use the shorthands $\mathsf{pred}(v) = \{v' \mid (v', v) \in E\}$ and $\mathsf{succ}(v) = \{v' \mid (v, v') \in E\}$. The *domain transition graph* $\mathit{DTG}(\Pi, v)$ of $v \in V$ is an arc-labeled digraph with nodes $\mathcal{D}(v)$ that contains an arc (ϑ, ϑ') labeled with $\mathsf{pre}(a) \setminus \mathsf{pre}(a)[v]$ iff $\mathsf{eff}(a)[v] = \vartheta'$ and either $\mathsf{pre}(a)[v] = \vartheta$ or $\mathsf{pre}(a)[v]$ is unspecified.

In this paper we extend two previously studied causal graph structures known as the *fork* and *inverted fork*. These structures are digraphs $G = (N, E)$ such that there exists a node $r \in N$ for which $(u, v) \in E \iff u = r$, if the structure is a fork, and $(u, v) \in E \iff v = r$, if the structure is an inverted fork. We refer to planning problems whose causal graphs are (inverted) forks as (inverted) fork structured planning problems. Optimal planning has been shown to be in P for fork structured planning problems if $|\mathcal{D}(r)| = 2$, and for inverted fork structured planning problems for any $|\mathcal{D}(r)| \in O(1)$ (Katz and Domshlak 2010).

Forks

We start by closing the gap left by Katz and Domshlak (2010) in the complexity of cost-optimal planning for fork-structured tasks:

Theorem 1 *Cost-optimal planning for fork structured problems with causal graph rooted in a ternary-valued variable is NP-complete.*

Proof: Membership in NP is obvious. The proof of hardness is by reduction from the shortest common superstring problem (SCS). Let $x_1, \ldots, x_n$ be a set of strings over a binary alphabet. Given x_i, let x'_i denote the string over the alphabet $\{0, 1, 2\}$ that results from inserting the symbol 2 at the beginning, end, and between each pair of symbols in x_i. There then exists an SCS of length k for $x_1, \ldots, x_n$ iff there exists an SCS of length $2k + 1$ for $x'_1, \ldots, x'_n$.

Given a planning problem $\Pi = \langle V, A, I, G, cost \rangle$, where:

- $V = \{r, y_1, \ldots, y_n\}$, with $\mathcal{D}(r) = \{0, 1, 2\}$ and $\mathcal{D}(y_i) = \{0, \ldots, |x'_i|\}$ for $i = 1, \ldots, n$,

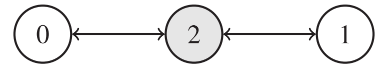

Figure 1: DTG for variable r.

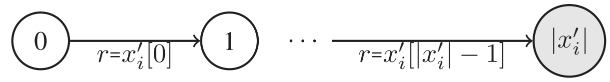

Figure 2: DTG for variable y_i.

- $A = \{a_{ij} \mid i = 1, \ldots, n, j = 0, \ldots, |x'_i| - 1\} \cup \{r_{0\to2}, r_{2\to0}, r_{1\to2}, r_{2\to1}\}$, where $a_{ij} = \langle \{y_i\text{=}j, r\text{=}x'_i[j]\}, \{y_i\text{=}j + 1\} \rangle$, in which $x'_i[j]$ denotes the jth symbol of x'_i, $r_{\alpha\to\beta} = \langle \{r\text{=}\alpha\}, \{r\text{=}\beta\} \rangle$, $cost(a_{ij}) = 0$ for all a_{ij} and $cost(r_{\alpha\to\beta}) = 1$,
- $I = \{r\text{=}2\} \cup \{y_i\text{=}0 \mid i = 1, \ldots, n\}$, and
- $G = \{r\text{=}2\} \cup \{y_i\text{=}|x'_i| \mid i = 1, \ldots n\}$,

finding an optimal plan for Π is equivalent to finding an SCS for $x'_1, \ldots, x'_n$. The causal graph of Π is a fork with root r and leaves $y_1, \ldots y_n$. The DTG for the variable r is a chain with 3 nodes, with the value 2 at the center doubly connected to each of the values 0, 1, at the two sides (Figure 1). The DTG for each of the variables $y_1, \ldots, y_n$ is a chain in which there is a single path that traverses the values of y_i in ascending order, and that requires for each transition that the variable r have the value corresponding to that position in the string x'_i (Figure 2).

Since the variables y_i can transition to their next values only when r has the value of the corresponding position in the string x'_i, the sequence of values taken on by the variable r must correspond to a superstring of the set of strings $\{x'_0, \ldots, x'_n\}$. The only actions with non-zero cost are those that change the value of r, and there therefore exists a plan for Π with cost $2k$ iff there exists a superstring of $\{x'_0, \ldots, x'_n\}$ with length $2k + 1$, and a superstring of $\{x_0, \ldots, x_n\}$ with length k. As this transformation can be performed in polynomial time, this shows the desired result. ■

Unfortunately, this does not shed light on the complexity of deciding plan existence. Our next result concerns this problem for fork-structured planning problems where a more general property holds for the DTG of the root variable:

Theorem 2 *Let Π be a planning task with a fork-structured causal graph rooted at variable r, and let $\mathcal{G}$ be the condensed graph of* $\mathit{DTG}(\Pi, r)$*, with one node for each strongly connected component (SCC) of* $\mathit{DTG}(\Pi, r)$*. Plan existence for Π can be decided in polynomial time if $\mathcal{G}$ has only a polynomial number of paths.*

Proof: Consider a (necessarily cycle-free, as the condensed graph is directed acyclic) path $P_1, \ldots, P_m$ in $\mathcal{G}$, where each node P_i corresponds to a *set* of values of r that make up an SCC in $\mathit{DTG}(\Pi, r)$. For $0 \leq i \leq m$ and for $v \in \mathsf{succ}(r)$, we define the sets C_v^i inductively as follows:

- $C_v^0 = \{I[v]\}$, and

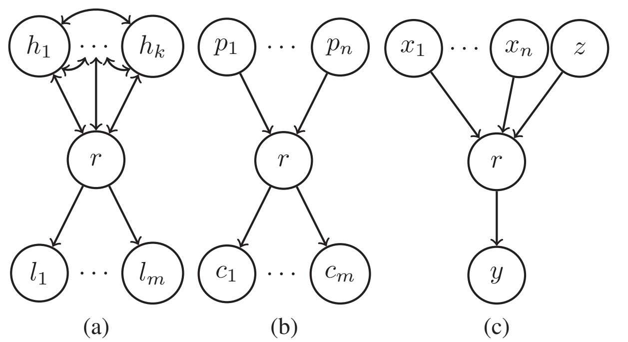

Figure 3: (a) Semifork and (b) hourglass causal graphs. (c) Causal graph structure for reduction of Theorem 7.

- for $i > 0$, C_v^i is the set of all values in $\mathcal{D}(v)$ achievable from any value in C_v^{i-1} using actions in A_v that have preconditions only on values of r that make up the SCC corresponding to P_i.

Note that it follows from this definition that C_v^i grows monotonically in i, i.e. $C_v^{i-1} \subseteq C_v^i$ for all i. Given a path $P_1, \ldots, P_m$ in $\mathcal{G}$, if for all $v \in \mathsf{succ}(r)$ we have $G[v] \in C_v^m$, and $G[r] \in P_m$, then a plan for Π can be constructed from the above in polynomial time. Π is solvable iff there exists a (cycle-free) path $P_1, \ldots, P_m$ in the condensed graph $\mathcal{G}$ such that $G[r] \in P_m$ and $G[v] \in C_v^m$ for all $v \in \mathsf{succ}(r)$. Since there are a polynomial number of paths to check, this proves the result. ■

We note that when $|\mathcal{D}(r)| = O(1)$, the condensed graph of ***DTG***(Π, r) has only $O(1)$ paths, and Theorem 2 is applicable. This result therefore implies that plan existence for fork-structured tasks with constant bounded root domains is in P and closes the gap left by Domshlak and Dinitz (2001).

Semifork Causal Graphs

We now explore a graph structure that we call a *semifork*:

Definition 1 (Semifork) *A digraph $G = (N, E)$ is a* semifork *if there exists a set of nodes $L \subset N$, $L \neq \emptyset$ such that (i) $\forall v \in L$ outdegree$(v) = 0$, and (ii) there exists a node $r \in N \setminus L$ such that $(u, v) \in E$ and $v \in L$ imply $u = r$.*

Informally, one part of a semifork causal graph has fork structure, and the remaining nodes have edges only among themselves or to the root of the fork (Figure 3a). We refer to the node r as the *center* of the causal graph, the nodes L as the semifork's *leaves*, and the rest of the nodes $N \setminus (L \cup \{r\})$ as the semifork's *hat*. Note that given a graph G, there may be multiple possibilities for choosing L that result in different interpretations of G as a semifork.[1] We now show a tractability result for semifork structured causal graphs, extending a previous result by Katz and Domshlak (2010):

[1] Each subset of the child nodes of a fork induces a different semifork when used as L, for example.

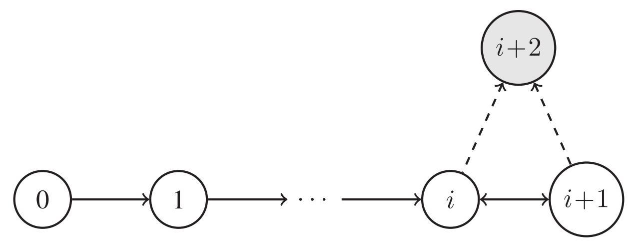

Figure 4: DTG for variable r_i in Π_{*i} (lower). Transitions represented with dashed edges may be present or not depending on the goal value defined for r or lack thereof.

Theorem 3 (Tractable Semiforks) *Given a constant k and a semifork-structured planning task $\Pi = \langle V, A, I, G, cost \rangle$ with center $r \in V$, $|\mathcal{D}(r)| = 2$, and $|\mathsf{hat}| < k$, cost-optimal planning for Π is polynomial in $\|\Pi\|^k$.*

Proof: We note that given a sequence of changes to r, the hat and fork portions of the planning problem can be decoupled and solved separately. Let Π^h denote the planning problem that results from removing all leaf variables from the problem, and $\pi(h)_i^*$ a cost minimal plan among the plans for Π^h in which the value of r is changed *at least* i times. In turn, let Π^f denote the problem in which all hat variables are removed and the value of r can be changed with no preconditions and cost 0, and $\pi(f)_i^*$ a cost minimal plan among the plans that set all the leaf variables to their goal values while changing the value of r *at most* i times. Any optimal plan π^* for Π can be partitioned into two such cost-minimal plans[2] by choosing i to be the number of changes to r in π^*. The optimal plan for Π can therefore be found by considering cost-minimal plans for Π^h and Π^f for each possible i:

$$cost(\pi^*(\Pi)) = \min_i [cost(\pi(h)_i^*) + cost(\pi(f)_i^*)]$$

and interleaving the actions of the two plans as required. Note that if (i) given a value of i, both $\pi(h)_i^*$ and $\pi(f)_i^*$ can be obtained in polynomial time, and (ii) there is an upper bound b on i that is polynomial in $\|\Pi\|$ such that both $cost(\pi(h)_i^*)$ and $cost(\pi(f)_i^*)$ are non-decreasing for $i > b$, the semifork problem can also be solved optimally in polynomial time. For $cost(\pi(h)_i^*)$, any bound will do, as increasing the value of i can only exclude plans making fewer changes to r. For $cost(\pi(f)_i^*)$, this bound is given by $b = \max_{v \in \mathsf{leaves}(r)} |\mathcal{D}(v)| + 1$ (Katz and Domshlak 2010). We now proceed to the formal description of how to obtain $\pi(h)_i^*$ and $\pi(f)_i^*$ in polynomial time.

We first describe the construction of a planning problem Π_i^h for $i \geq 1$, whose optimal plans correspond to optimal plans $\pi(h)_i^*$. Assuming wlog that $I[r] = 0$, we restrict Π to the variables $\mathsf{hat} \cup \{r\}$, while modifying the DTG of r to consist of $i + 3$ values (Figure 4):

- $V_i = \mathsf{hat} \cup \{r_i\}$, with $\mathcal{D}(r_i) = \{0, \ldots, i + 2\}$
- $$A_i = \bigcup_{v \in \mathsf{hat}} A_v \cup \bigcup_{j=0}^{i+1} A^j \cup A^g$$

[2] Otherwise, each could be independently replaced with any cost-minimal plan.

where $A^g = \{a^g_i, a^g_{i+1}\}$ if no goal value is defined for r, $A^g = \{a^g_i\}$ if $G[r] + i$ is even, and $A^g = \{a^g_{i+1}\}$ if $G[r] + i$ is odd, where $a^g_j = \langle \{r_i{=}j\}, \{r_i{=}i+2\} \rangle$. For $0 \leq j \leq i$,

$$A^j = \bigcup_{a \in A_r} \left\{ a_f \;\middle|\; \begin{array}{l} \mathsf{pre}(a_f)[r_i] = j, \mathsf{eff}(a_f)[r_i] = j+1, \\ \mathsf{pre}(a)[r] + j \text{ is even, and} \\ \mathsf{pre}(a_f)[v] = \mathsf{pre}(a)[v] \text{ and} \\ \mathsf{eff}(a_f)[v] = \mathsf{eff}(a)[v] \quad \forall v \in \mathsf{hat} \end{array} \right\},$$

and for $j = i + 1$,

$$A^{i+1} = \bigcup_{a \in A_r} \left\{ a_b \;\middle|\; \begin{array}{l} \mathsf{pre}(a_b)[r_i] = i+1, \mathsf{eff}(a_b)[r_i] = i, \\ \mathsf{pre}(a)[r] + i + 1 \text{ is even, and} \\ \mathsf{pre}(a_b)[v] = \mathsf{pre}(a)[v] \text{ and} \\ \mathsf{eff}(a_b)[v] = \mathsf{eff}(a)[v] \quad \forall v \in \mathsf{hat} \end{array} \right\},$$

and $cost_i(a_f) = cost(a)$, $cost_i(a_b) = cost(a)$, $cost_i(a^i_g) = cost_i(a^{i+1}_g) = 0$,

- $I_i[v] = I[v]$ for $v \in \mathsf{hat}(r)$ and $I_i[r_i] = 0$, and
- $G_i[v] = G[v]$ for $v \in \mathsf{hat}(r)$ and $G_i[r_i] = i + 2$.

Note that due to the requirement that $\mathsf{pre}(a)[r]{+}j$ be even, actions preconditioned by $r{=}0$ appear in A^j only for even j and those preconditioned by 1 for odd j. In order to reach the goal value of r_i, the plan must apply a sequence of actions that change r i times, and can then alternate between the values i and $i+1$ before achieving the goal, preconditioned on the original goal value of r. Since the task Π^h_i has at most k variables, it is solvable optimally in polynomial time, and a cost-minimal plan $\pi(h)^*_i$ can be obtained by replacing the actions in an optimal plan for Π^h_i with the corresponding actions from A, that is, replacing r_i-changing actions with their r-changing originals.

We now consider how to obtain the plans $\pi(f)^*_i$. Given a sequence of value changes of the variable r, all children $c_j \in \mathsf{leaves}(r)$ are independent of each other and of the hat. Provided a number i of value changes for r, a cost-minimal plan for each child variable can therefore be obtained in polynomial time, and these plans can be interleaved to obtain a cost minimal plan.[3]

In order to obtain an optimal plan for Π, it is therefore sufficient to iterate over all values $0 \leq i \leq b$, where $b = \max_{v \in \mathsf{leaves}(r)} |\mathcal{D}(v)| + 1$, and store the plans that result in the cheapest summed cost $\pi(h)^*_i + \pi(f)^*_i$. These plans can then be interleaved by adding the actions in $\pi(f)^*_i$ at the earliest possible point during the execution of $\pi(h)^*_i$ to obtain an optimal plan. ■

Relaxing the constant bound on the size of hat makes even the plan existence problem NP-complete, as arbitrary planning problems can then be encoded. The same is the case when the binary bound on the domain size of the center variable is relaxed:

[3]For further detail see the proof of Theorem 4 by Katz and Domshlak (2010).

Theorem 4 *Plan existence for semifork structured problems with $|\mathsf{hat}| = 1$ and center variable domain size ≥ 3 is NP-complete.*

Proof: The idea behind the proof is similar to that of Theorem 1. Given a set of strings over a binary alphabet and a parameter k, we construct a planning problem in the same way as we did there, except with an additional variable x on which all actions that change the value of r have a prevail condition. The causal graph of this problem is then a semifork with a single variable in the hat. The domain transition graph of x is a chain of length $2k$, alternating values of which allow transitions in r from 0 or 1 to 2 and from 2 to 0 or 1, respectively. This variable enforces that the value of r can be changed from 2 to either 0 or 1 and then back to 2 at most k times, and as before the problem is then solvable iff there exists a superstring of the set of strings of length k. ■

Hourglass Causal Graphs

We now introduce a digraph structure that we call the *hourglass* (Figure 3b):

Definition 2 (Hourglass) *A digraph $G = (N, E)$ is an hourglass if (i) $(u, v) \in E$ implies $(v, u) \notin E$, and (ii) there exists a node $r \in N$, such that for each $(u, v) \in E$, either $u = r$ or $v = r$.*

We call the node r the *center* of the graph. We refer to its predecessor nodes $\mathsf{pred}(r) = \{u \in N \mid (u, r) \in E\}$ as *parents*, and its successor nodes $\mathsf{succ}(r) = \{v \in N \mid (r, v) \in E\}$ as *children*. Intuitively, the hourglass differs from the semifork in that edges between the parent nodes are not allowed, and the outgoing edges of the center node all lead to child variables. We begin with the positive result that imposing a constant domain bound on the center and the child variables makes optimal planning tractable:

Theorem 5 (Hourglass with bound on child domain size) *Given a constant d and an hourglass-structured planning task Π with center variable domain size $|\mathcal{D}(r)| \leq d$, and $|\mathcal{D}(c_i)| \leq d$ for all child variables $c_i \in \mathsf{succ}(r)$, optimal planning for Π is polynomial in $\|\Pi\|^k$, where $k = d^{(d^2+2)}$.*

Proof: First, we note that the bound d on the domain size of the child variables also constitutes a bound on the length of the sequence of prevail values required from r for any one child. Considering also that up to d intermediate values of r may be required in moving from one value to another, the total length of the sequence of r values for a single child is d^2. The number of all possible sequences of that length is d^{d^2}, and a (loose) upper bound on the length of a sequence that contains all such sequences as subsequences is given by $k = d^2 \cdot d^{d^2} = d^{(d^2+2)}$. The number of possible r-changing action sequences that can achieve these values is then a polynomial $|A|^k$. Given such an action sequence, an optimal sequence of actions for the parent variables that satisfies all the required preconditions can be found in linear time. It is therefore sufficient to check each possible sequence of actions up to length k and choose the one that results in the globally optimal plan. ■

However, when such a bound is not imposed, even satisficing planning quickly becomes NP-complete:

Theorem 6 *Satisficing planning for hourglasses with center variable domain size ≥ 3 is NP-complete.*

This follows trivially from the proof of Theorem 4, as the problem in the proof has hourglass structure. Bounding the domain sizes of the child variables without bounding that of the center variable does not help either, as it follows from results for inverted forks by Domshlak and Dinitz (2001) that satisficing planning in this case is NP-complete.

We now consider the complexity of planning for problems with hourglass causal graphs with the added parameter of k-*dependence* (Katz and Domshlak 2008):

Definition 3 (k-dependent) *An action a is k-dependent if the size of its prevail condition, that is the number of variables that it has preconditions on but whose values it does not change, is $\leq k$. A planning problem Π is k-dependent if all its actions are k-dependent.*

We first show that for 2-dependent hourglass-structured problems even satisficing planning is NP-complete:

Theorem 7 (2-dependent hourglass) *Plan existence for the 2-dependent hourglass problem with center variable domain size 2 is NP-complete.*

Proof: Membership in NP is obvious, we show hardness by a polynomial reduction from SAT. Let $P = (C, U)$ be a SAT problem with m clauses $C = \{C_0, \ldots, C_{m-1}\}$ and n variables $U = \{u_1, \ldots, u_n\}$. We construct an hourglass problem Π with a single child variable y and $n+1$ parent variables $x_1, \ldots, x_n, z$ (Figure 3c). The goal of the problem is defined only for the child variable y, and its purpose is to force the value of the center variable to change exactly $2m-1$ times. Its DTG is therefore an ascending chain of length $2m$ with values $0, \ldots, 2m-1$, transitions $i \to i+1$ that require alternating values of r beginning with r=1, and goal $2m-1$. A solution to Π then exists iff the parent nodes of the problem permit the value of r to be changed $2m-1$ times. The parent variables $x_1, \ldots, x_n$ correspond to the variables of the SAT problem, and have DTGs that allow their values to be set once to either 0 or 1, from an initial "undefined" value. The variable z has a DTG which consists of a chain with $2m$ values, whose even values $2i$, in conjunction with a value for some variable appearing in C_i that satisfies it, allow r to be set to 1, and whose odd values allow r to be set to 0. To solve the problem, a plan must set the values of the x_i variables to appropriate values, and advance through the DTG of Z while setting alternating values for r.

Formally, we define $\Pi = \langle V, A, I, G, cost\rangle$ as follows:

- $V = \{x_1, \ldots, x_n, z, r, y\}$
- $I = \{r{=}0, x_1{=}\perp, \ldots, x_n{=}\perp, z{=}0, y{=}0\}$
- $G = \{y{=}2m-1\}$
- $$A = \bigcup_{i=1}^{n} A_{x_i} \cup \bigcup_{i=0}^{2m-1} \{a_y^i\} \cup \bigcup_{i=0}^{2m-3} \{a_z^i\} \cup A_{r\to 0} \cup \bigcup_{i=0}^{m-1} A_{r\to 1}^i$$

  where
  - $A_{x_i} = \{\langle\{x_i{=}\perp\}, \{x_i{=}0\}\rangle, \langle\{x_i{=}\perp\}, \{x_i{=}1\}\rangle\}$,
  - $a_y^i = \langle\{r{=}(i \mod 2), y{=}i\}, \{y{=}i+1\}\rangle$,
  - $a_z^i = \langle\{z{=}i\}, \{z{=}i+1\}\rangle$,
  - $A_{r\to 0} = \bigcup_{i=1}^{2m-3}\{\langle\{r{=}1, z{=}i\}, \{r{=}0\}\rangle \mid i \text{ is odd}\}$, and
  - $A_{r\to 1}^i = \bigcup_{u_j=\theta\in C_i}\{\langle\{r{=}0, z{=}2i, x_j{=}\theta\}, \{r{=}1\}\rangle\}$.

Note that the largest k-dependence in Π is 2. As pointed out above, a solution for Π exists iff the value of r can be changed $2m-1$ times, and the value of r can be changed $2m-1$ times iff there exists an assignment that satisfies clauses $C_0, \ldots, C_{m-1}$. As the initial value of r is 0, $2m-1$ changes of r indicates that r must change from 0 to 1 m times. Each of these changes must be caused by actions that are drawn from the sets $A_{r\to 1}^i$ for different values of i, since each consecutive change to r depends on different values of z. Due to the construction of the set of actions $A_{r\to 1}^i$, an action from this set can be applied iff C_i is satisfied. Therefore plan existence implies that all C_i are satisfied. ■

We now consider the 1-dependent case, first proving a lemma that leads to our tractability result for hourglasses:

Lemma 1 (Optimal plans for 1-dependent Hourglasses) *Given an hourglass-structured 1-dependent planning problem Π with $|\mathcal{D}(r)| = 2$, there exists an optimal plan for Π in which the actions changing the value of r have prevail conditions on at most two variables.*

Proof: Let π^* be an optimal plan for Π, and let π_r^* be the subsequence of π^* consisting only of actions in A_r. For each $\theta \in \mathcal{D}(r)$, let $a_\theta^* = \mathrm{argmin}_{a\in\pi_r^*}\{cost(a) \mid \mathsf{eff}(a)[r] = \theta\}$, and let x and x' be the two variables on which the actions a_θ^* for $\theta \in \mathcal{D}(r)$ have prevail conditions. If $x \neq x'$, then it is possible to construct from π^* a new optimal plan π'^* that uses only these cheapest actions to change the value of r. The remaining actions that change the values of other parent variables or those of the child variables can be left unchanged. Since the actions we replace are no cheaper than a_θ^*, the result is also an optimal plan. Note that this also holds if one or both of the cheapest actions have no prevail conditions.

For the more complicated case in which $x = x'$, let

$$(a', \theta') = \underset{a\in\pi_r^*, \theta\in\mathcal{D}(r)}{\mathrm{argmin}} \left\{ \begin{array}{c|c} cost(a)+ & \mathsf{eff}(a)[r] \neq \theta \wedge \\ cost(a_\theta^*) & \mathsf{pre}(a)[x] \text{ is unspecified} \end{array} \right\}$$

In words, a' is an action not prevailed by x that together with $a_{\theta'}^*$ gives the lowest summed cost for two actions changing r from one value to another and back, at least one of which is not prevailed by x. If such an action does not exist, then π_r^* complies with the above property. We now show how to obtain an optimal plan for Π in which all of the r-changing actions are either prevailed by x or are occurrences of a'. Let a_1, a_2 denote two consecutive actions in π_r^* such that at least one of a_1, a_2 is not prevailed by x, and $\{a_1, a_2\} \neq \{a', a_{\theta'}^*\}$. If no such pair of consecutive actions exists, then the condition described above is met. Otherwise, we construct a new sequence $\pi_r'^*$ by inserting in π_r^* immediately after the first occurence of $a_{\theta'}^*$ the two actions $a', a_{\theta'}^*$, and removing the two actions a_1, a_2. As noted earlier, the summed cost of a' and $a_{\theta'}^*$ is minimal among two r-changing actions at least one of which is not prevailed by x, and $\pi_r'^*$ is therefore no more expensive than π_r^*. Since the value that prevails a' and the sequence of distinct values of x that prevail actions in $\pi_r'^*$ are achieved by π_r^*, a new plan can be constructed by scheduling the actions achieving these values appropriately with respect to the actions in $\pi_r'^*$. As above, actions affecting other parent variables and child variables can be left untouched. The result is an optimal plan π'^* that complies with the above property. ■

Lemma 1 allows us to concentrate on optimal plans of a certain structure, and therefore solve this type of hourglass problem optimally in polynomial time:

Theorem 8 *Optimal planning for* 1*-dependent hourglasses with center variable domain size* $|\mathcal{D}(r)| = 2$ *is in* P.

Proof: For each subset V' of size 2 of the parents $\mathsf{pred}(r)$ we create a planning problem Π' by removing from Π all r-changing actions that have preconditions on the variables in $\mathsf{pred}(r) \setminus V'$. From lemma 1 we have that an optimal plan for one such problem Π' is also an optimal plan for our original problem Π. Since Π' consists of the set of singleton variables $\mathsf{pred}(r) \setminus V'$, each of which can be solved in polynomial time, along with the rest of the problem which is a semifork with a hat of size 2, which can also be solved in polynomial time (Theorem 3), Π' can also be solved optimally in polynomial time. Since the number of Π' problems that must be considered to find an optimal plan for Π is polynomial, optimal planning for Π is in P. ■

Finally, note that the planning problem in the proof of Theorem 6 is 1-dependent, as the indegree of each state variable is bounded by 1. Even satisficing planning for 1-dependent hourglasses with $|\mathcal{D}(r)| \geq 3$ is therefore NP-hard, completing our complexity map of the hourglass fragment.

Practice

Our tractability results for cost-optimal planning suggest that implicit abstraction heuristics can be made more informative. A semifork with a single hat variable, for example, can naturally represent fuel constraints for a mobile in transportation domains (Helmert 2008). However, there are a number of issues which must be attended to before the semifork and hourglass patterns can be employed in the framework of structural pattern database heuristics. Given a planning task Π over the variables V and a variable $v \in V$, the first issue is how to select a semifork or hourglass centered at v. For a constant k bounding the size of the hat and a

$\lvert\mathcal{D}(r)\rvert$	2	3	$O(1)$	$\Theta(n)$
F	P/—	NPC/—	—/P	—/NPC
SF	P/—	—/NPC		
$H_{\lvert Ch\rvert=O(1)}$			P/—	—/NPC
$H(1)$	P/—	—/NPC		
$H(2)$	—/NPC			

Figure 5: Complexity of cost-optimal/satisficing planning for **F**orks, **S**emi**F**orks with constant bound on hat size, and **H**ourglasses, with k-dependence in parentheses. "—" and empty columns indicate that the complexity is implied by other results. Results implied by previous work are shaded.

set of variables $V' \subseteq (V \setminus \{v\})$ of size $\leq k$, a semifork with hat V' can be constructed by dropping all outgoing edges from $L = V \setminus (V' \cup \{v\})$ and all edges from $V \setminus \{v\}$ to L, leaving only edges from v to L and among $V' \cup \{v\}$. As the number of such sets V' is polynomial in k, all possible such semiforks could be accounted for. Hourglasses are more problematic, as when there exists $v' \in V$ such that edges (v, v') and (v', v) are both in *CG*(Π), there is a choice of whether to use v' as a child or a parent. The second issue is how to abstract the problem to have G as its causal graph, modifying the set of actions to be consistent with its edges. For hourglasses, the previously defined acyclic causal-graph decomposition (Katz and Domshlak 2010) can be used, but must be adapted to account for possible cycles in semiforks. The last issue is that the chosen set of abstractions must be efficiently solvable in the states encountered during search. For this, most of the calculations can be performed prior to search and cached. The choice of the values to be precalculated and stored remains a subject of research.

Conclusions

We have extended the analysis of the complexity of planning problems described in terms of the structure of the causal graph, k-dependence, and the domain sizes of variables (Figure 5). We have closed some gaps left open in previous work, showing that optimal planning for fork causal graphs with root variable domain size ≥ 3 is NP-complete, and that satisficing planning is in P for arbitrary constant sized domains. We have introduced new causal graph fragments, called the semifork and hourglass, that generalize the previously known fork and inverted fork structures. Optimal planning for semiforks with center variable domain size 2 and a constant bound on the number of variables in the hat turns out to be in P, as does optimal planning for hourglasses with binary center variable domain and k-dependence 1. Relaxing the bound on domain size in either case results in a problem that is NP-complete even for satisficing planning, and the same is true of relaxing the bound on k-dependence for hourglasses. A number of questions must be addressed before these patterns can be used in the framework of structural pattern database heuristics.

Acknowledgments. Work performed while Michael Katz was employed by INRIA, Nancy, France, supported by the

French National Research Agency (ANR), project ANR-10-CEXC-003-01.

References

Bäckström, C., and Nebel, B. 1995. Complexity results for SAS$^+$ planning. *Computational Intelligence* 11(4):625–655.

Brafman, R. I., and Domshlak, C. 2003. Structure and complexity of planning with unary operators. *Journal of Artificial Intelligence Research* 18:315–349.

Brafman, R. I., and Domshlak, C. 2006. Factored planning: How, when, and when not. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI)*, 809–814.

Chen, H., and Giménez, O. 2008. Causal graphs and structurally restricted planning. In *Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS)*, 36–43.

Domshlak, C., and Dinitz, Y. 2001. Multi-agent off-line coordination: Structure and complexity. In *Proceedings of Sixth European Conference on Planning (ECP)*, 277–288.

Giménez, O., and Jonsson, A. 2008. The complexity of planning problems with simple causal graphs. *Journal of Artificial Intelligence Research* 31:319–351.

Giménez, O., and Jonsson, A. 2009. The influence of k-dependence on the complexity of planning. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*, 138–145.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*, 162–169.

Helmert, M. 2004. A planning heuristic based on causal graph analysis. In *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS)*, 161–170.

Helmert, M. 2008. *Understanding Planning Tasks: Domain Complexity and Heuristic Decomposition*, volume 4929 of *Lecture Notes in Computer Science*. Springer.

Katz, M., and Domshlak, C. 2008. New islands of tractability of cost-optimal planning. *Journal of Artificial Intelligence Research* 32:203–288.

Katz, M., and Domshlak, C. 2010. Implicit abstraction heuristics. *Journal of Artificial Intelligence Research* 39:51 – 126.

Knoblock, C. 1994. Automatically generating abstractions for planning. *Artificial Intelligence* 68(2):243–302.