Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling

Optimal Search with Inadmissible Heuristics

Erez Karpas and **Carmel Domshlak**
Faculty of Industrial Engineering & Management,
Technion, Israel

Abstract

Considering cost-optimal heuristic search, we introduce the notion of *global admissibility* of a heuristic, a property weaker than standard admissibility, yet sufficient for guaranteeing solution optimality within forward search. We describe a concrete approach for creating globally admissible heuristics for domain independent planning; it is based on exploiting information gradually gathered by the search via a new form of reasoning about what we call *existential optimal-plan landmarks*. We evaluate our approach on some state-of-the-art heuristic search tools for cost-optimal planning, and discuss the results of this evaluation.

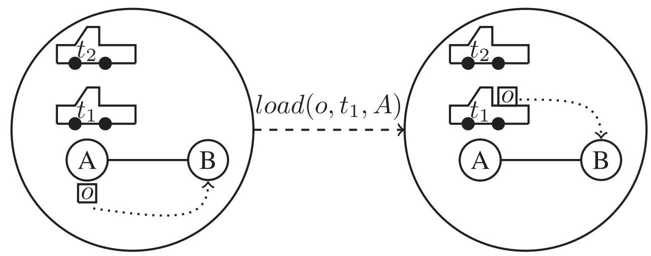

Figure 1: Example Logistics task

Introduction

These days, the most prominent domain-independent approach to cost-optimal deterministic planning is state-space search with the A^* algorithm and admissible heuristic functions (Hart, Nilsson, and Raphael 1968). A heuristic function is admissible if it never overestimates the cost of achieving the goal from the given state. Recent progress in cost-optimal planning is primarily due to spectacular advances in automatic construction of admissible heuristics (Bonet and Geffner 2001; Haslum and Geffner 2000; Edelkamp 2001; Helmert, Haslum, and Hoffmann 2007; Katz and Domshlak 2010a; Karpas and Domshlak 2009; Helmert and Domshlak 2009). Admissibility, however, is also an unfortunately strong property: adopting admissibility may force the search to examine an exponential number of states even if the heuristic is almost perfect (Pearl 1984; Helmert and Röger 2008).

In our work we revive a long-standing observation that, at least in theory, heuristic admissibility is not a necessary condition for forward search to guarantee optimality of the discovered plan (Dechter and Pearl 1985). We define a weaker yet sufficient condition of *global admissibility*, and introduce a concrete inference technique that yields such globally admissible heuristics. Our technique reasons about candidate plan prefixes π generated by the search process, utilizing the well-known notion of causal links (Tate 1977). Causal links are widely exploited in partial-order planning (Penberthy and Weld 1992; Mcallester and Rosenblitt 1991), constraint-based planning (Vidal and Geffner 2006), and recently also in satisficing state space search (Lipovetzky and Geffner 2011). Our use of causal links here is novel: we use them to infer constraints that must be satisfied by an optimal plan having π as its prefix, and then use these constraints to enhance the heuristic evaluation of the end-state of π.

The technique is based on the simple observation that, for each action along each optimal plan for the problem, there must be some justification for applying that action. Consider a simple logistics problem, depicted in Figure 1, with two locations A and B, two trucks t_1 and t_2, and a single package o. In the initial state both trucks and the package are at location A, and the goal is to have the package at B. Clearly, any solution must load the package into one of the trucks. Now, suppose we have already loaded the package onto truck t_1. While it is still possible to unload the package from t_1, load it onto t_2, and use t_2 to deliver the package to B, any *optimal* solution from the state in question will exploit the fact that some effort has already been put into loading the package onto t_1, and will use t_1 to deliver the package. This is precisely the type of inference our technique attempts to perform.

We note that our work is in fact not the first in the practice of cost-optimal planning to rely on global admissibility. Optimality-preserving techniques for search space pruning, such as symmetry breaking and state-space reductions (Fox and Long 2002; Rintanen 2003; Coles and Smith 2008; Chen and Yao 2009; Pochter, Zohar, and Rosenschein 2011;

Domshlak, Katz, and Shleyfman 2012), can be seen as assigning a heuristic value of ∞ to some states, despite the fact that the goal is achievable from these states. In that respect, our work can be seen as extending the palette of techniques, as well as (and even more importantly) the sources of information that can be used for relaxing admissibility: while the aforementioned pruning techniques are typically based on syntactic properties of the problem description (such as functional equivalence of two objects), the technique described in what follows performs a continuous semantic analysis of information revealed by the search process.

Bounded intention planning (Wolfe and Russell 2011) is based upon a similar notion to ours, that each action should have some purpose. However, the way we exploit this notion is quite different from BIP. Specifically, we do not modify the planning task, and our technique is not restricted to unary-effect problems with acyclic causal graphs.

Preliminaries

We consider planning tasks formulated in STRIPS with action costs; our notation mostly follows that of Helmert and Domshlak (2009). A planning task $\Pi = \langle P, A, \mathcal{C}, s_0, G \rangle$, where P is a set of propositions, A is a set of actions, each of which is a triple $a = \langle \mathsf{pre}(a), \mathsf{add}(a), \mathsf{del}(a) \rangle$, $\mathcal{C} : A \to \mathbb{R}^{0+}$ is a cost function on actions, $s_0 \subseteq P$ is the initial state, and $G \subseteq P$ is the goal. For ease of notation and without loss of generality, in what follows we assume that there is a single goal proposition ($G = \{p_g\}$), which can only be achieved by a single action, END.

An action a is applicable in state s if $\mathsf{pre}(a) \subseteq s$, and if applied in s, results in the state $s' = (s \setminus \mathsf{del}(a)) \cup \mathsf{add}(a)$. A sequence of actions $\langle a_0, a_1, \ldots, a_n \rangle$ is applicable in state s_0 if a_0 is applicable in s_0 and results in state s_1, a_1 is applicable in s_1 and results in s_2, and so on. The cost of action sequence $\pi = \langle a_0, a_1, \ldots, a_n \rangle$ is $\sum_{i=0}^{n} \mathcal{C}(a_i)$, and is denoted by $\mathcal{C}(\pi)$. The state resulting from applying action sequence π in state s is denoted by $s[\![\pi]\!]$. If π_1 and π_2 are action sequences, by $\pi_1 \cdot \pi_2$ we denote the concatenation of π_1 and π_2. An action sequence $\langle a_0, a_1, \ldots, a_n \rangle$ is an s-path if it is applicable in state s, and it is also an s-plan if $a_n = $ END. Optimal plans for Π are its cheapest s_0-plans, and the objective of cost-optimal planning is to find such an optimal plan for Π. We denote the cost of a cheapest s-plan by $h^*(s)$.

Finally, let $\pi = \langle a_0, a_1, \ldots a_n \rangle$ be an action sequence applicable in state s. The triple $\langle a_i, p, a_j \rangle$ forms a *causal link* in π if $i < j$, $p \in \mathsf{add}(a_i)$, $p \in \mathsf{pre}(a_j)$, $p \notin s[\![\langle a_0, a_1, \ldots a_{i-1} \rangle]\!]$, and for $i < k < j$, $p \notin \mathsf{del}(a_k) \cup \mathsf{add}(a_k)$. In other words, a_i is the actual provider of precondition p for a_j. In such a causal link, a_i is called the *provider*, and a_j is called the *consumer*.

Minimal Preconditions and Intended Effects

Before we describe our inference technique in detail, we must first define some basic notions, upon which our inference technique is based. The first such notion is that of *minimal precondition* after following path π, which refers to the minimal set of propositions that is needed to continue π into an optimal plan. However, the formal notion is somewhat involved, and distinguishes between the minimal preconditions after an s_0-path π in the context of different sets of optimal plans:

Definition 1 (Minimal Precondition)
Let $\Pi = \langle P, A, \mathcal{C}, s_0, G \rangle$ be a planning task, and OPT *be a set of optimal plans for Π. Given an s_0-path $\pi = \langle a_0, a_1, \ldots a_n \rangle$, a set of propositions $X \subseteq s_0[\![\pi]\!]$ is an* OPT-minimal precondition after π *iff there exists an X-plan π' such that $\pi \cdot \pi' \in$* OPT *and for every $X' \subset X$, π' is not applicable in X'.*

In other words, X is an OPT-minimal precondition after π if it is possible to continue π into some plan in OPT, using only the propositions in X. While Definition 1 is intuitive, our inference technique is better understood by considering the notion of the *intended effects* of an s_0-path π:

Definition 2 (Intended Effect)
Let $\Pi = \langle P, A, \mathcal{C}, s_0, G \rangle$ be a planning task, and OPT *be a set of optimal plans for Π. Given an s_0-path $\pi = \langle a_0, a_1, \ldots a_n \rangle$, a set of propositions $X \subseteq s_0[\![\pi]\!]$ is an* OPT-intended effect of π *iff there exists an $s_0[\![\pi]\!]$-plan π' such that $\pi \cdot \pi' \in$* OPT *and π' consumes exactly X, that is, $p \in X$ iff there is a causal link $\langle a_i, p, a_j \rangle$ in $\pi \cdot \pi'$, with $a_i \in \pi$ and $a_j \in \pi'$.*

The basic observation underlying this notion is very simple, if not to say trivial: every action along an optimal plan should be there for a reason — there should be some use of at least one of the effects of each of the plan's actions. The following theorem shows that the notions of intended effects and minimal preconditions are equivalent:

Theorem 1 (Definition Equivalence)
Let $\Pi = \langle P, A, \mathcal{C}, s_0, G \rangle$ be a planning task with $s_0 = \emptyset$, and a unique START *action. Then Definitions 1 and 2 are equivalent.*

The proof of Theorem 1 can be found in Appendix A. As mentioned previously, our inference technique is better understood by considering intended effects, and thus we will continue and discuss only those. We denote the set of all OPT-intended effects of an s_0-path π by *IE*$(\pi|\text{OPT})$; when OPT is the set of *all* optimal plans for Π, then *IE*$(\pi|\text{OPT})$ is simply called "intended effects" and is denoted by *IE*(π). Note that if π is not a cheapest path from s_0 to $s_0[\![\pi]\!]$ then *IE*$(\pi|\text{OPT}) = \emptyset$ for all optimal plan sets OPT.

We illustrate this concept using the logistics task depicted in Figure 1. There are two optimal solutions for this task: one using truck t_1 to deliver the package, and another using truck t_2. Thus, assuming the initial state is established by a START action, it is easy to see that the intended effects of the initial state are described by *IE*$(\langle \text{START} \rangle) = \{\{at(t_1, A), at(o, A)\}, \{at(t_2, A), at(o, A)\}\}$. However, after loading the package o into truck t_1, there is only one optimal way to continue — by delivering the package using t_1, and thus *IE*$(\langle \text{START}, load(o, t_1, A) \rangle) = \{\{at(t_1, A), in(o, t_1)\}\}$.

If provided to us, the intended effects of π can reveal valuable information about what any continuation of π must do. For example, if for some proposition p we have $p \in X$ for all intended effects $X \in \mathit{IE}(\pi)$, then clearly any optimal continuation of π must contain some action consuming p. This example suggests that intended effects of π can be used either for deriving a heuristic estimate of $s_0[\![\pi]\!]$, or for enhancing such an existing estimate. We now suggest one such framework for exploiting intended effects. It is based on what we call *existential optimal-plan landmarks*, or *$\exists$-opt landmarks*, for short.

First, interpreting proposition subsets $X \subseteq P$ as valuations of P, assume that a set of intended effects $\mathit{IE}(\pi|\text{OPT})$ is given to us as a propositional logic formula ϕ such that that $X \in \mathit{IE}(\pi|\text{OPT}) \Leftrightarrow X \models \phi$. By $M(\phi)$ we denote the set of ϕ's models, that is, $M(\phi) = \{X \subseteq P \mid X \models \phi\}$. For an s_0-path π, let us also treat any continuation π' of π as a valuation of P, assigning *true* to the propositions produced by π *and* consumed by π', and *false* to all other propositions. This way, the semantics of statements "π' satisfies ϕ", $\pi' \models \phi$, is well defined. In our example logistics task, such a formula ϕ for the intended effects of the START action could be $\phi = at(o, A) \wedge \big(at(t_1, A) \vee at(t_2, A)\big)$. After applying $load(o, t_1, A)$, the intended effects are described by $at(t_1, A) \wedge in(o, t_1)$.

Theorem 2
Let OPT *be a set of optimal plans for a planning task* Π*,* π *be an* s_0*-path, and* ϕ *be a propositional logic formula describing* $\mathit{IE}(\pi|\text{OPT})$*. Then, for any* $s_0[\![\pi]\!]$*-plan* π'*,* $\pi \cdot \pi' \in \text{OPT}$ *implies* $\pi' \models \phi$*.*

Theorem 2, proof of which is immediate from Definition 2, establishes our interpretation of the formula ϕ as a $\exists$-opt landmark: While ϕ is not a landmark in the standard sense of this term (Hoffmann, Porteous, and Sebastia 2004), that is, not every plan (and not even every optimal plan) must satisfy ϕ, *some optimal* plan starting with π must satisfy ϕ after π.

In line with the recent work on regular landmarks, henceforth we assume that ϕ is given in CNF. The CNF representation of ϕ is advantageous mainly in that it has a natural interpretation as a set of disjunctive fact landmarks, where each clause describes one such landmark. Note that unlike regular landmarks, where a fact landmark stands for a disjunctive action landmark composed of its *achievers*, in our $\exists$-opt landmarks a fact stands for a disjunctive action landmark composed of its *consumers.* However, it is possible, for instance, to combine the information captured by the $\exists$-opt landmark(s) ϕ and the information captured by the regular landmarks of the h_{LA} heuristic, by performing an action cost partition over the union of their landmarks (Karpas and Domshlak 2009). When cost partitioning is optimized via, e.g., the linear programming technique (Karpas and Domshlak 2009; Katz and Domshlak 2010b), the resulting estimate is guaranteed to dominate h_{LA}. In fact, if we could find just a single OPT-intended effect $X \in \mathit{IE}(\pi|\text{OPT})$, we could then use X as a *regular* landmark, pruning some parts of the search space without sacrificing optimality: Since we know there must exist some continuation π' with $\pi \cdot \pi' \in \text{OPT}$, X by itself constitutes an $\exists$-opt landmark.

So far, we have outlined the promise of $\exists$-opt landmarks induced by intended effects, yet that promise is, of course, only potential since the intended effects $\mathit{IE}(\pi|\text{OPT})$ were assumed to be somehow provided to us. It is hardly surprising, however, that finding just a single intended effect of an action sequence is as hard as STRIPS planning itself.

Theorem 3
Let INTENDED be the following decision problem: Given a planning task $\Pi = \langle P, A, \mathcal{C}, s_0, G\rangle$*, an* s_0*-path* π*, and a set of propositions* $X \subseteq P$*, is* $X \in \mathit{IE}(\pi)$*?*
Deciding INTENDED is PSPACE*-hard.*

Proof: The proof is by reduction from the complement of PLANSAT — the problem of deciding whether a given planning task is solvable. For STRIPS, PLANSAT is known to be PSPACE-hard even when all actions are unit cost (Bylander 1994), and since PSPACE = CO-PSPACE, so is its complement.

Given a planning task $\Pi = \langle P, A, \mathcal{C}, s_0, G\rangle$ with unit cost actions and $|P| = n$, we construct a new planning task $\Pi' = \langle P', A', \mathcal{C}', s_0', G'\rangle$ as follows:

- $P' := P \cup \{d_i \mid 0 \leq i \leq n+1\}$;
- $A' := A \cup \{\text{inc}(i) \mid 0 \leq i \leq n+1\}$, where $\text{inc}(i) = \langle\{d_j \mid j < i\}, \{d_i\}, \{d_j \mid j < i\}\rangle$;
- $\mathcal{C}'$ assigns costs of 1 to all actions in a';
- $s_0' := s_0$; and
- $G' := G \vee (d_0 \wedge d_1 \wedge \ldots \wedge d_{n+1})$.

The goal G' is disjunctive, but this disjunction can be straightforwardly compiled away.

Note that Π' is always solvable because there will always be a solution of cost $2^{n+2} - 1$, using the inc operators to increment a binary counter composed of $d_0, \ldots, d_{n+1}$. Now, if the original task Π is solvable, then it has a solution of cost at most $2^n - 1$. Therefore, the inc operators and the d_i propositions are part of an optimal solution iff Π is not solvable, and thus $\{d_0\}$ is an optimal intended effect in Π' after applying $\text{inc}(0)$ iff Π is not solvable. ■

Although Theorem 3 shows that computing $\mathit{IE}(\pi|\text{OPT})$ precisely is not feasible, the promise of $\exists$-opt landmarks still remains: we can approximate $\mathit{IE}(\pi|\text{OPT})$ while still guaranteeing optimality, and thus maintain the correctness of the reasoning. In particular, below we show that any superset of $\mathit{IE}(\pi|\text{OPT})$ induces *possible intended effects* and provides such a "safe" approximation.

Theorem 4
Let OPT *be a set of optimal plans for a planning task* Π*,* π *be an* s_0*-path,* $\mathit{PIE}(\pi|\text{OPT}) \supseteq \mathit{IE}(\pi|\text{OPT})$ *be a set of possible* OPT*-intended effects of* π*, and* ϕ *be a logical formula describing* $\mathit{PIE}(\pi|\text{OPT})$*. Then, for any* $s_0[\![\pi]\!]$*-plan* π'*,* $\pi \cdot \pi' \in \text{OPT}$ *implies* $\pi' \models \phi$*.*

Proof: Let π' be an $s_0[\![\pi]\!]$-plan such that $\pi \cdot \pi' \in$ OPT, and let X be the set of all propositions produced by π and consumed by π'. From Definition 2, $X \in IE(\pi|\text{OPT})$, and since $IE(\pi|\text{OPT}) \subseteq PIE(\pi|\text{OPT})$, $X \in PIE(\pi|\text{OPT})$. Since ϕ describes $PIE(\pi|\text{OPT})$, it holds that $X \models \phi$. ∎

We now proceed with describing a concrete proposal for finding and utilizing useful *PIE* approximations of this type in the context of OPT containing either all optimal plans or just one optimal plan.

Approximating Intended Effects

One easy way of obtaining a set $PIE(\pi)$ such that $IE(\pi) \subseteq PIE(\pi)$ is to take $PIE(\pi) = 2^P$. Needless to say, however, it provides us with no useful information whatsoever. A slightly tighter approximation of $IE(\pi)$ would be $PIE(\pi) = 2^{s_0[\![\pi]\!]}$. Clearly, no continuation of π can consume anything that π achieved but does not hold in the state reached by π. However, this approximation of $IE(\pi)$ still does not provide us with any useful information.

We begin by showing that it is possible to obtain a much tighter approximation of $IE(\pi|\text{OPT})$ for OPT consisting of a single optimal plan ρ, and that this approximation can provide us with useful information about the OPT-continuations of π. Obviously, the plan ρ will not actually be known to us; or otherwise there would be no point in planning in the first place. In itself, however, that will not be an obstacle.

This approximation is based on exploiting a set of s_0-paths, which we refer to as out shortcut library $\mathcal{L}$; later we discuss how such a library can be obtained automatically, but for now we assume we are simply provided with one. Given a planning task Π, let $\prec$ be a lexicographic order on its action sequences, based on an arbitrary total order of the actions. Let ρ be the optimal plan for Π that is minimal (lowest) with respect to $\prec$. That is the plan we focus on, and we can now describe our approximation of $IE(\pi|\{\rho\})$. For that, we define an additional ordering $\lhd$ on action sequences:

$$\pi' \lhd \pi \Leftrightarrow \mathcal{C}(\pi') < \mathcal{C}(\pi) \vee (\mathcal{C}(\pi') = \mathcal{C}(\pi) \wedge \pi' \prec \pi).$$

Note that $\pi' \lhd \pi$ implies $\mathcal{C}(\pi') \leq \mathcal{C}(\pi)$. Using $\lhd$, we approximate $IE(\pi|\{\rho\})$ with

$$PIE_{\mathcal{L}}(\pi|\{\rho\}) = \{X \subseteq s_0[\![\pi]\!] \mid \nexists \pi' \in \mathcal{L} : \pi' \lhd \pi, X \subseteq s_0[\![\pi']\!]\}.$$

In other words, for any $X \subseteq s_0[\![\pi]\!]$, $\mathcal{L}$ *proves* that X is *not* an intended effect of π if it can offer a cheaper way of achieving X from s_0, or if it can offer a way of achieving X from s_0 at the same cost, but that alternative way is "preferred" to π with respect to $\prec$.

Theorem 5
Let $\prec$ be a lexicographic order on action sequences, and let ρ be an optimal solution of Π that is minimal in $\prec$. For any s_0-path π, it holds that $IE(\pi|\{\rho\}) \subseteq PIE_{\mathcal{L}}(\pi|\{\rho\})$.

Proof: Assume to the contrary that there exists some $X \in IE(\pi|\{\rho\}) \setminus PIE_{\mathcal{L}}(\pi|\{\rho\})$. Since $X \in IE(\pi|\{\rho\})$, there exists some path π' such that $\pi \cdot \pi' = \rho$, and π' consumes all propositions in X. $X \notin PIE_{\mathcal{L}}(\pi|\{\rho\})$, so from the definition of $PIE_{\mathcal{L}}(\pi|\{\rho\})$ there exists some s_0-path $\pi'' \in \mathcal{L}$ such that π'' is applicable at s_0, $\pi'' \lhd \pi$, and $X \subseteq s_0[\![\pi'']\!]$.

π' is applicable at $s_0[\![\pi'']\!]$, since π' consumes exactly the propositions in X, and $X \subseteq s_0[\![\pi'']\!]$. $\pi \cdot \pi' = \rho$ is a valid plan, so the last action in π' must be the END action, which implies that $\pi'' \cdot \pi'$ is a valid plan.

$\pi'' \lhd \pi$, and so one of (I-II) below must be true:

I $\mathcal{C}(\pi'') < \mathcal{C}(\pi)$.
But then, $\mathcal{C}(\pi'' \cdot \pi') < \mathcal{C}(\pi \cdot \pi')$, contradicting the optimality of $\pi \cdot \pi' = \rho$.

II $\mathcal{C}(\pi'') = \mathcal{C}(\pi)$ and $\pi'' \prec \pi$.
But then $\mathcal{C}(\pi'' \cdot \pi') = \mathcal{C}(\pi \cdot \pi')$, and thus $\pi'' \cdot \pi'$ is an optimal plan. Since $\prec$ is a lexicographic order, $\pi'' \cdot \pi' \prec \pi \cdot \pi'$, contradicting the minimality of $\pi \cdot \pi' = \rho$ in $\prec$.

We have seen that either case of $\pi'' \lhd \pi$ leads to a contradiction, thus proving the theorem. ∎

We note that, very similarly, one can obtain an approximation of the intended effects $IE(\pi)$ of π with respect to *all* optimal plans, with no need for the lexicographic order $\prec$ on the action sequences. Specifically, one can use

$$PIE_{\mathcal{L}}(\pi) = \{X \subseteq s_0[\![\pi]\!] \mid \nexists \pi' \in \mathcal{L} : \mathcal{C}(\pi') < \mathcal{C}(\pi), X \subseteq s_0[\![\pi']\!]\},$$

and the proof that $IE(\pi) \subseteq PIE_{\mathcal{L}}(\pi)$ is very similar to the proof of Theorem 5. However, it is worth discussing the differences between the two approximations. Using landmarks derived from $PIE_{\mathcal{L}}(\pi)$ will yield admissible estimates along *all* optimal plans, while using landmarks derived from $PIE_{\mathcal{L}}(\pi|\{\rho\})$ might prune some, possibly almost all, optimal plans. However, $PIE_{\mathcal{L}}(\pi|\{\rho\})$ is guaranteed to never prune ρ, ensuring that there will always be at least one optimal plan along which estimates are admissible, namely ρ. We get back to this point later on when we discuss the modifications that need to be made to A^* to guarantee optimality of the search.

From $PIE_{\mathcal{L}}(\pi|\{\rho\})$ to Optimal-Plan Landmarks

As mentioned, in order to use $PIE_{\mathcal{L}}(\pi|\{\rho\})$ as a set of disjunctive fact landmarks, we have to derive a CNF formula that compactly represents it. Recall that $PIE_{\mathcal{L}}(\pi|\{\rho\})$ consists of all sets of propositions in $s_0[\![\pi]\!]$ for which there is no "shortcut" in $\mathcal{L}$. Let π be an s_0-path, and let $\pi' \in \mathcal{L}$ be an s_0-path in the library such that $\pi' \lhd \pi$. If π is the beginning of our $\prec$-minimal optimal plan ρ, then there must be some proposition p consumed by the continuation of π along ρ which is achieved by π but not by π', that is, $p \in s_0[\![\pi]\!] \setminus s_0[\![\pi']\!]$. In CNF, this information, derived from π on the basis of the library $\mathcal{L}$, is encoded as

$$\phi_{\mathcal{L}}(\pi|\{\rho\}) = \bigwedge_{\pi' \in \mathcal{L} : \pi' \lhd \pi} \vee_{p \in s_0[\![\pi]\!] \setminus s_0[\![\pi']\!]} \; p.$$

As a special case of $\phi_{\mathcal{L}}(\pi|\{\rho\})$, note that if there exists $\pi' \in \mathcal{L}$ with $\mathcal{C}(\pi') < \mathcal{C}(\pi)$ and $s_0[\![\pi]\!] \subseteq s_0[\![\pi']\!]$, then $\phi_{\mathcal{L}}(\pi|\{\rho\})$ contains an empty clause, meaning that there is no optimal continuation of π. This is the case captured by the definition of a dominating action sequence (Nedunuri, Cook, and Smith 2011), and $\phi_{\mathcal{L}}(\pi|\{\rho\})$ generalizes their definition. The following theorem demonstrates that $\phi_{\mathcal{L}}(\pi|\{\rho\})$ describes *PIE*$_{\mathcal{L}}(\pi|\{\rho\})$, and thus by Theorem 5 approximates *IE*$(\pi|\{\rho\})$:

Theorem 6
For any s_0-path π, PIE$_{\mathcal{L}}(\pi|\{\rho\}) = M(\phi_{\mathcal{L}}(\pi|\{\rho\}))$.

Proof: We first show that *PIE*$_{\mathcal{L}}(\pi|\{\rho\}) \subseteq M(\phi_{\mathcal{L}}(\pi|\{\rho\}))$. Assume to the contrary that there exists $X \in$ *PIE*$_{\mathcal{L}}(\pi|\{\rho\}) \setminus M(\phi_{\mathcal{L}}(\pi|\{\rho\}))$. $X \notin M(\phi_{\mathcal{L}}(\pi|\{\rho\}))$, so there exists some clause $c_{\pi'} = \vee_{p \in s_0[\![\pi]\!] \setminus s_0[\![\pi']\!]} p$ in $\phi_{\mathcal{L}}(\pi|\{\rho\})$, corresponding to path $\pi' \in \mathcal{L}$, which X does not satisfy. Since the clause $c_{\pi'}$ contains the propositions $s_0[\![\pi]\!] \setminus s_0[\![\pi']\!]$, this implies that $X \cap \big(s_0[\![\pi]\!] \setminus s_0[\![\pi']\!]\big) = \emptyset$. We know that $X \subseteq s_0[\![\pi]\!]$, and so we have that $X \subseteq s_0[\![\pi']\!]$. However, $X \in$ *PIE*$_{\mathcal{L}}(\pi|\{\rho\})$, so there is no "shortcut" in $\mathcal{L}$ for achieving X. Therefore, for any $\pi' \in \mathcal{L}$ such that $\pi' \lhd \pi$, we have that $X \not\subseteq s_0[\![\pi']\!]$—a contradiction.

We now show that $M(\phi_{\mathcal{L}}(\pi|\{\rho\})) \subseteq$ *PIE*$_{\mathcal{L}}(\pi|\{\rho\})$. Assume to the contrary that there exists $X \in M(\phi_{\mathcal{L}}(\pi|\{\rho\})) \setminus$ *PIE*$_{\mathcal{L}}(\pi|\{\rho\})$. $X \notin$ *PIE*$_{\mathcal{L}}(\pi|\{\rho\})$, so there exists some $\pi' \in \mathcal{L}$ such that $\pi' \lhd \pi$ and $X \subseteq s_0[\![\pi']\!]$. Let $c_{\pi'} = \vee_{p \in s_0[\![\pi]\!] \setminus s_0[\![\pi']\!]} p$ be the clause corresponding to π' in $\phi_{\mathcal{L}}(\pi|\{\rho\})$. We know that $X \subseteq M(\phi_{\mathcal{L}}(\pi|\{\rho\}))$, thus X must satisfy every clause of $\phi_{\mathcal{L}}(\pi|\{\rho\})$, and specifically, X must satisfy $c_{\pi'}$. However, $X \subseteq s_0[\![\pi']\!]$, and $c_{\pi'}$ does not contain any of these propositions. Thus, X cannot satisfy $c_{\pi'}$ — a contradiction. ■

Similarly, we can derive a CNF formula $\phi_{\mathcal{L}}(\pi)$ which describes *PIE*(π) (with a very similar proof of the equivalence):

$$\phi_{\mathcal{L}}(\pi) = \bigwedge_{\pi' \in \mathcal{L} : \mathcal{C}(\pi') < \mathcal{C}(\pi)} \vee_{p \in s_0[\![\pi]\!] \setminus s_0[\![\pi']\!]} \; p.$$

Obtaining a Shortcut Library

So far we assumed our "shortcut" library $\mathcal{L}$ is given. We now describe a concrete approach to obtaining it. Importantly, note that $\mathcal{L}$ does not have to be a static list of action sequences, but rather can be generated dynamically for each s_0-path π constructed by the search procedure. In particular, such a path-specific library can be generated using a set of rules similar to the plan rewrite rules of Nedunuri, Cook, and Smith (2011). A plan rewrite rule is a rule of the form $\pi_1 \rightarrow \pi_2$, where π_1 and π_2 are some action sequences and the rule means that *whenever* π_1 is a subsequence of a plan, it can be replaced in that plan with π_2 without violating the plan's validity. We do not require such a strong connection between π_1 and π_2. First, instead of requiring that π_1 be applicable whenever π_2 is applicable, we can simply check whether π_1 is applicable for the current state. Second, we do not require π_2 to achieve everything that π_1 does, since we can also exploit information from the set difference of their effects — that is, that any optimal continutation will need to use something that π_1 achieved and π_2 did not.

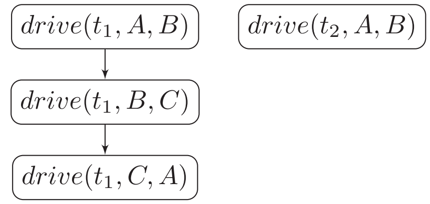

Figure 2: Causal Structure of Example

Given a path π and a set of plan rewrite rules, we construct a library $\mathcal{L}(\pi)$ specifically for π. To construct $\mathcal{L}(\pi)$, we first need to define a digraph, which we call the *causal structure of* π. The nodes of this graph are the action instances in π, and there is an edge from a_i to a_j when there is a causal link in π with a_i as the provider and a_j as the consumer. Given a plan rewrite rule $\pi_1 \rightarrow \pi_2$ with $\pi_2 \lhd \pi_1$, we can check whether π_1 appears as a chain in the causal structure of π. We can then attempt to replace π_1 with π_2, and check if the resulting action sequence is still applicable in s_0. Note that π_1 does not have to appear in π as a contiguous subsequence, making this a more general strategy than simply looking for contiguous occurrences of π_1.

Our current implementation is a special case of this scheme that uses plan rewrite rules of the form $\pi \rightarrow \langle\rangle$, that is, the tail of each rule is the empty sequence. We look for two types of chains in the causal structure of π. The first type are isolated chains, that is, chains with no edges going out from any node in the chain to any node outside the chain. This ensures that removing every suffix of the causal chain still leads to a valid s_0-path. As a special case of this, removing the last operator in each causal chain will yield the same landmarks as those from the "unjustified actions" (Karpas and Domshlak 2011), up to ordering of the action sequence. The second type of chains we look for correspond to "action a supports its inverse action a'", where the notion of action invertibility is adopted from Hoffmann (2002).

To illustrate how this process works, we consider the following example action sequence in a Logistics problem: $\pi = \langle drive(t_1, A, B),\ drive(t_1, B, C),\ drive(t_2, A, B),\ drive(t_1, C, A)\rangle$. The causal structure of this action sequence is shown in Figure 2. Clearly, truck t_1 drives in a loop here, without doing anything useful on the way. Thus, $\langle drive(t_1, A, B),\ drive(t_1, B, C),\ drive(t_1, C, A)\rangle$ form an isolated causal chain. We can replace this causal chain with the empty sequence, yielding the "shortcut" $\pi' = \langle drive(t_2, A, B)\rangle$, which leads to the same state as π, allowing us to prove that π cannot be the beginning of any optimal solution. Note that while truck t_1 drives in a loop, there is no loop in the state space (which could be detected by the search algorithm), since truck t_2 moved in between the moves of t_1. However, our shortcut library allows us to eliminate non-contiguous subsequences of π, making this stronger than simple duplicate state detection.

Unfortunately, as the following example Blocksworld task demonstrates, our inference technique is an *approximation* of the intended effects. Assume in the initial state, the crane is holding block A, and the goal is to have all of the blocks on the table. Then performing $putdown(A)$ adds three facts: $ontable(A), clear(A)$ and $hand\text{-}empty$. $ontable(A)$ is a goal, and is an intended effect of $\langle putdown(A)\rangle$. If there are other blocks that are out of place, meaning that the crane needs to move more blocks, $hand\text{-}empty$ is also an intended effect. However, $clear(A)$ is not an intended effect of $\langle putdown(A)\rangle$, as we do not need to move block A, or put any other blocks on it. Nevertheless, our inference technique can only deduce that we need to use one of the effects of $putdown(A)$, thus generating the $\exists$-opt landmark $ontable(A) \vee clear(A) \vee hand\text{-}empty$, which has $clear(A)$ as a possible intended effect.

In terms of previous uses of causal links, the PROBE planner's (Lipovetzky and Geffner 2011) causal commitments bear similarity to our use of the causal structure, with one major difference. While PROBE chooses the causal commitments (that is, which proposition each action *will* be used to achieve) as soon as the action is chosen, we look at all possible causal commitments. This is why PROBE cannot guarantee optimality, while we can.

Optimality Without Admissibility

Performing some admissible action cost partitioning over $\exists$-opt landmarks is not guaranteed to yield an admissible heuristic. This is because the $\exists$-opt landmarks we derive are not guaranteed to hold in every possible plan, but rather only in *some optimal* plans. However, admissibility is a strong requirement, which states that the heuristic h must not overestimate the distance from *every* state to the goal. We now define a weaker notion than admissibility, which we call *global admissibility*.

Definition 3 (Globally Admissible Heuristics)
A heuristic h is called globally admissible *if, for any planning task Π, if Π is solvable, then there exists an optimal plan ρ for Π such that, for any state s along ρ, $h(s) \leq h^*(s)$.*

Following the proof that A^* leads to finding an optimal solution (Pearl 1984, p. 78), it is easy to see that the same proof also works when h is globally admissible. The intuition behind this is that if h is globally admissible, then it is admissible for the states along some optimal plan, and thus these states will be expanded by A^*. Given that, the heuristic estimates can be arbitrarily high for all other states as we anyway prefer not to expand them. As a special case, a globally admissible heuristic can assign a value of ∞ to a state which is not on the "chosen" optimal solution, declaring it as a dead-end. And while this sufficiency of global admissibility has been well noted before (Dechter and Pearl 1985), to the best of our knowledge (and somewhat to our surprise), no formal definition of this property for a heuristic appears in the literature.

Having said that, it should be noted that incorporating our $\exists$-opt landmarks into heuristic estimate of a state s makes the resulting heuristic path-dependent, because our $\exists$-opt landmarks come from reasoning about a concrete s_0-path π to s. A path-dependent heuristic assigns a heuristic value to a path, rather than a state. As, in principle, a path-dependent estimate can depend only on the end-state of the path, path-dependent heuristics generalize the more common state-dependent heuristics.

We therefore define a new type of admissibility for path-dependent heuristics, which, like intended effects is based on a set of plans.

Definition 4 (χ-path Admissible Heuristics)
*Let χ be a set of valid plans for Π. Path-dependent heuristic h is χ-*path admissible *if, for any planning task Π, for any prefix π of any plan $\rho \in \chi$, $h(\pi) \leq h^*(s_0[\![\pi]\!])$.*

A χ-path admissible heuristic assigns admissible estimates to any prefix of any plan in χ. Note that χ-path admissible heuristics are not "admissible" in the traditional sense, as they might assign a non-admissible estimate to some *state*, if the path used to evaluate that state is not a prefix of some plan in χ. If χ is the set of all optimal plans, and h is χ-path admissible, we will simply call h path admissible.

For a set of optimal plans $\textsc{opt}$, it is fairly easy to see that given a CNF formula $\phi_{\mathcal{L}}(\pi|\textsc{opt})$ which describes some approximation of $IE(\pi|\textsc{opt})$, any admissible action cost partition over $\phi_{\mathcal{L}}(\pi|\textsc{opt})$ yields an $\textsc{opt}$-path admissible heuristic. Specifically, any admissible action cost partition over $\phi_{\mathcal{L}}(\pi|\{\rho\})$ yields a $\{\rho\}$-path admissible heuristic, and any admissible action cost partition over $\phi_{\mathcal{L}}(\pi)$ yields a path admissible heuristic.

However, using A^* with a path admissible heuristic does not guarantee optimality of the solution found. For example, suppose there is only a single optimal solution, but one of the states along that optimal solution is first reached via a sub-optimal path. Then the heuristic value associated with that state could be ∞, and the optimal solution will be pruned.

Given a path admissible heuristic, such as the one derived from $\phi_{\mathcal{L}}(\pi)$, we can guarantee finding an optimal solution. To do this, we modify A^* to recompute the heuristic estimate for a state s every time a cheaper path to s has been found. Note that if a new path to s, that is more expensive than the currently best-known path, is found, then the heuristic estimate derived from that path is clearly not guaranteed to be admissible, as admissibility is only guaranteed for prefixes of some optimal solution.

For an arbitrary χ-path admissible heuristic, it is not as easy to guarantee that an optimal solution will be found. However, where our specific inference technique for $\phi_{\mathcal{L}}(\pi|\{\rho\})$ is concerned, we know which optimal solution is the "chosen" solution: the $\prec$-minimal optimal plan ρ. Thus we recompute the heuristic estimate for s when a cheaper path to s is found, or if a path of the same cost as the best known path, which is lower according to $\prec$ is found. These two small modifications to A^* are enough to guarantee the optimality of the solution that is found, even when the heuristic in use is not admissible.

Empirical Evaluation

Although the direction of using $\exists$-opt landmarks to derive a χ-path admissible heuristic is interesting, *a priori* it is not

coverage	$\phi_{\mathcal{L}}(\pi)$	$\phi_{\mathcal{L}}(\pi\|\{\rho\})$	h_{LA}	*LM-A**
airport (50)	**28**	27	**28**	28
blocks (35)	21	21	21	21
depot (22)	**5**	**5**	4	4
driverlog (20)	**9**	**9**	7	7
elevators (30)	**7**	0	**7**	7
freecell (80)	**51**	49	**51**	51
grid (5)	2	2	2	2
gripper (20)	5	5	5	5
logistics00 (28)	20	20	20	20
logistics98 (35)	3	3	3	3
miconic (150)	141	141	141	141
mprime (35)	**19**	17	15	15
mystery (30)	**15**	**15**	12	12
openstacks (30)	12	12	12	12
parcprinter (30)	**12**	**12**	11	11
pathways (30)	4	4	4	4
pegsol (30)	26	26	26	26
pipesworld-notankage (50)	15	15	15	15
pipesworld-tankage (50)	**10**	8	**10**	9
psr-small (50)	48	48	48	48
rovers (40)	5	5	5	5
satellite (36)	**6**	4	4	4
scanalyzer (30)	13	13	13	13
sokoban (30)	**15**	0	**15**	15
storage (30)	**14**	13	13	13
tpp (30)	**6**	5	5	5
transport (30)	9	9	9	9
trucks-strips (30)	**7**	**7**	6	6
woodworking (30)	11	11	11	11
zenotravel (20)	8	8	8	8
SUM	**547**	514	531	530

Table 1: Coverage

expansions	$\phi_{\mathcal{L}}(\pi)$	$\phi_{\mathcal{L}}(\pi\|\{\rho\})$	h_{LA}
airport (27)	**211052**	420947	211647
blocks (21)	**1064433**	1160581	1070441
depot (4)	**290141**	388822	401696
driverlog (7)	**170534**	224226	363541
freecell (49)	**403030**	556692	**403030**
grid (2)	**227288**	231599	467078
gripper (5)	**458498**	594875	**458498**
logistics00 (20)	**816589**	1487932	862443
logistics98 (3)	**13227**	22014	45654
miconic (141)	**135213**	183319	**135213**
mprime (15)	**35308**	42093	313576
mystery (14)	**37698**	48785	290133
openstacks (12)	**1579931**	1756117	**1579931**
parcprinter (11)	**101178**	146959	158090
pathways (4)	**32287**	58912	173593
pegsol (26)	**3948303**	4364821	**3948303**
pipesworld-notankage (15)	**1248036**	1775363	1377390
pipesworld-tankage (8)	**24080**	36830	28761
psr-small (48)	**358647**	373242	698003
rovers (5)	**98118**	343152	231380
satellite (4)	**5906**	8817	10623
scanalyzer (13)	**22251**	27893	23213
storage (13)	**313259**	359482	475049
tpp (5)	**4227**	7355	12355
transport (9)	**915027**	1062859	929285
trucks-strips (6)	**230699**	314618	1261745
woodworking (11)	**92195**	163589	152975
zenotravel (8)	**66600**	86782	186334
SUM	**12903755**	16248676	16269980

Table 2: Expansions

clear how much we can gain from following this direction in practice. Therefore, we have implemented the ∃-opt landmarks machinery on top of the Fast Downward planning system (Helmert 2006), and performed an empirical evaluation, comparing the regular landmarks heuristic h_{LA} (Karpas and Domshlak 2009) using the complete set of delete-relaxation fact landmarks (Keyder, Richter, and Helmert 2010) to the same heuristic enhanced with ∃-opt landmarks. In order to avoid loss of accuracy due to differences in action cost partitioning, we used only the optimal cost partitioning, computed via linear programming (Karpas and Domshlak 2009). The evaluation comprised all the IPC 1998-2008 STRIPS domains, including those with non-uniform action costs. All of the experiments reported here were run on a single core of an Intel E8400 CPU, with a time limit of 30 minutes and a memory limit of 6 GB, on a 64-bit linux OS.

We tested the two variants of ∃-opt landmark formulae discussed above: $\phi_{\mathcal{L}}(\pi)$ and $\phi_{\mathcal{L}}(\pi|\{\rho\})$. Recall that while $\phi_{\mathcal{L}}(\pi)$ is path admissible, $\phi_{\mathcal{L}}(\pi|\{\rho\})$ is $\{\rho\}$-path admissible. Consequently, the criterion for when to recompute the heuristic value for a state which is reached via a new path differs: With $\phi_{\mathcal{L}}(\pi)$ we only recompute the heuristic estimate for state s when a cheaper path to s has been found. With $\phi_{\mathcal{L}}(\pi|\{\rho\})$ we also recompute the heuristic estimate when the new path π' to s of the same cost as the current path π has been found, if π' is lexicographically lower than π.

Table 1 shows the number of problems solved in each domain under each configuration. The column titled h_{LA} lists the number of problems solved using the regular landmarks heuristic, and the columns titled $\phi_{\mathcal{L}}(\pi)$ and $\phi_{\mathcal{L}}(\pi|\{\rho\})$ list the number of problems solved in each domain using the respective ∃-opt landmarks. Because *A** is not suited for the multi-path dependent landmark heuristic h_{LA} (Karpas and Domshlak 2009), for h_{LA} we used the same search algorithm as for $\phi_{\mathcal{L}}(\pi)$, which recomputes the heuristic estimate for a state when a cheaper path to it is found. As the results show, using $\phi_{\mathcal{L}}(\pi)$ solves the most problems, because $\phi_{\mathcal{L}}(\pi)$ is more informative than h_{LA} alone. The poor performance of $\phi_{\mathcal{L}}(\pi|\{\rho\})$ can be explained by its pruning of optimal solutions those other than ρ, which increases the time until a solution is found.

Table 2 lists the total number of expansions performed in each domain, on the problems solved by all three configurations. Note that the ELEVATORS and SOKOBAN domains are missing from this table, as using $\phi_{\mathcal{L}}(\pi|\{\rho\})$ results in solving no problems in these two domains. The results clearly demonstrate that $\phi_{\mathcal{L}}(\pi)$ greatly increases the informativeness over h_{LA}.

Having said that, using ∃-opt landmarks limits our choice of search algorithm, and specifically prevents us from using *LM-A** (Karpas and Domshlak 2009) — a search algorithm specifically designed to exploit the multi-path dependence of h_{LA}. We therefore also compared to the number of problems solved by *LM-A** using the h_{LA} heuristic. The number of problems this configuration solved appears in Table 1, under *LM-A**. As the results show, our modified version of *A** with $\phi_{\mathcal{L}}(\pi)$ solves more problems than *LM-A** with h_{LA}. We do not compare the number of states expanded by our modified *A** with ∃-opt landmarks with these expanded by *LM-A**: comparing the number of expansions between different search algorithms does not tell us anything about the informativeness of the heuristics used.

Conclusion and Future Work

We have defined the notions of global admissibility and χ-path admissibility, and demonstrated that it is possible to derive a χ-path admissible heuristic and exploit it in cost-optimal planning. Our experimental results indicate that this technique can substantially reduce the number of states that must be expanded until an optimal solution is found.

While the heuristic we evaluated in this paper is already more informative than the regular landmarks heuristic, we believe this is not the end of the road. First, the dynamic shortcut library generation process can be improved by introducing more general forms of plan rewrite rules — not just rules which attempt to delete some action sequences from the current path, but rules which attempt to *replace* some action sequences with other action sequences. There are several possible sources for these rules, including learning them online, during search.

Furthermore, it is quite likely that other methods of deriving $\exists$-opt landmarks could be found. In fact, the inference technique we present here could be enhanced with additional reasoning, as demonstrated in the following scenario. Assume that action a was applied, and achieved proposition p. Our current inference technique can deduce that at some later point, some action which consumes p must be applied. Still, the question is *when* a consumer of p should be applied. One natural option is to apply it directly after a. However, there are two possible reasons this might not be the best choice: either the consumer requires some other preconditions which have not yet been achieved, or the consumer threatens another action, which should be applied before the consumer. Incorporating this type of reasoning into our inference technique poses an interesting challenge.

Acknowledgements

This work was carried out in and supported by the Technion-Microsoft Electronic-Commerce Research Center. We thank Malte Helmert for discussions and helpful advice on early drafts of this paper.

Appendix A

Theorem 1 (Definition Equivalence)
Let $\Pi = \langle P, A, \mathcal{C}, s_0, G\rangle$ be a planning task with $s_0 = \emptyset$, and a unique START *action. Then Definitions 1 and 2 are equivalent.*

Proof: Let X be a OPT-minimal precondition after π. Then there exists some path π', such that $\pi \cdot \pi' \in$ OPT, and π' is applicable in X, but not in any proper subset of X.

To show that X is an OPT-intended effect of π, we must show that $p \in X$ iff there exists some causal link $\langle a_i, p, a_j\rangle$ in $\pi \cdot \pi'$, with $a_i \in \pi$ and $a_j \in \pi'$. Denote the achiever of $p \in s_0[\![\pi]\!]$ by $ach(p) := \max_{\{i|0 \leq i \leq n, p \in \mathsf{add}(a_i)\}} i$. Every proposition must have an achiever, because $s_0 = \emptyset$.

If $p \in X$, then it must be a precondition of some action in π', because otherwise π' would be applicable in $X \setminus \{p\}$. Denote the first action in π' which has p as a precondition by $cons(p)$. Then clearly $\langle ach(p), p, cons(p)\rangle$ is a causal link as required.

If $p \notin X$, then either (a) there is no action in π' which has p as a precondition, or (b) there is some action $a_j \in \pi'$ which has p as a precondition, and there is some action $a_i \in \pi'$ with $i < j$ which achieves p. Otherwise, π' would not be an X-plan. In case (a), there is clearly no causal link on p with a consumer in π', as there is no action in π' which requires p. In case (b) denote the first action in π' which requires p by a_j, and denote by $a_i \in \pi'$ the latest action to achieve p before a_j. There is no causal link on p with a producer in π.

We have seen that $p \in X$ iff there exists some causal link $\langle a_i, p, a_j\rangle$ in $\pi \cdot \pi'$, with $a_i \in \pi$ and $a_j \in \pi'$.

Now assume X is an OPT-intended effect of π. Then there exists some path π', such that $p \in X$ iff there exists some causal link $\langle a_i, p, a_j\rangle$ in $\pi \cdot \pi'$, with $a_i \in \pi$ and $a_j \in \pi'$. We will show that X is a OPT-minimal precondition after π (that is, that π' is applicable in X, but not in any proper subset of X).

Assume to the contrary that π' is not applicable in X. Denote by a_j the first action in π' which is not applicable, and denote by $p \in \mathsf{pre}(a_j)$ some proposition that does not hold before applying a_j (after following π' until a_j from state X). If $p \in X$, then there is some causal link $\langle a_i, p, a_j\rangle$ in $\pi \cdot \pi'$, with $a_i \in \pi$ and $a_j \in \pi'$. But then p could not have been deleted before a_j, and $p \in X$, which means that p must hold before applying a_j — a contradiction. If $p \notin X$, then there is no causal link between π and π' on p. Therefore, when applying π' in $s_0[\![\pi]\!]$, p must be achieved by some action in π'. But then, when applying π' in X, a_j should be applicable — a contradiction.

Therefore, π' is applicable in X. We must now show that there is no $X' \subset X$, such that π' is applicable in X'. Assume to the contrary that there exists such X', and let $p \in X \setminus X'$. $p \in X$, so there must exist some causal link $\langle a_i, p, a_j\rangle$ in $\pi \cdot \pi'$, with $a_i \in \pi$ and $a_j \in \pi'$. π' is applicable in X', but but $p \notin X'$, implying that some action in π' achieved p for a_j. But $\langle a_i, p, a_j\rangle$ is a causal link in $\pi \cdot \pi'$, with $a_i \in \pi$, implying that there is no action that achieves p before a_j in π' — a contradiction.

■

References

2002. *Proc. AIPS 2002*.

Bonet, B., and Geffner, H. 2001. Planning as heuristic search. *AIJ* 129(1):5–33.

Bylander, T. 1994. The computational complexity of propositional STRIPS planning. *AIJ* 69(1–2):165–204.

Chen, Y., and Yao, G. 2009. Completeness and optimality preserving reduction for planning. In *Proc. IJCAI 2009* (2009), 1659–1664.

Coles, A. I., and Smith, A. J. 2008. Upwards: The role of analysis in cost optimal SAS+ planning. In *IPC-2008 planner abstracts*.

Dechter, R., and Pearl, J. 1985. Generalized best-first search strategies and the optimality of A*. *JACM* 32(3):505–536.

Domshlak, C.; Katz, M.; and Shleyfman, A. 2012. Enhanced symmetry breaking in cost-optimal planning as forward search. In *Proc. ICAPS 2012*.

Edelkamp, S. 2001. Planning with pattern databases. In *Proc. ECP 2001*, 13–24.

Fox, M., and Long, D. 2002. Extending the exploitation of symmetries in planning. In *Proc. AIPS 2002* (2002), 83–91.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2):100–107.

Haslum, P., and Geffner, H. 2000. Admissible heuristics for optimal planning. In *Proc. AIPS 2000*, 140–149.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In *Proc. ICAPS 2009*, 162–169.

Helmert, M., and Röger, G. 2008. How good is almost perfect? In *Proc. AAAI 2008*, 944–949.

Helmert, M.; Haslum, P.; and Hoffmann, J. 2007. Flexible abstraction heuristics for optimal sequential planning. In *Proc. ICAPS 2007*, 176–183.

Helmert, M. 2006. The Fast Downward planning system. *JAIR* 26:191–246.

Hoffmann, J.; Porteous, J.; and Sebastia, L. 2004. Ordered landmarks in planning. *JAIR* 22:215–278.

Hoffmann, J. 2002. Local search topology in planning benchmarks: A theoretical analysis. In *Proc. AIPS 2002* (2002), 92–100.

2009. *Proc. IJCAI 2009*.

Karpas, E., and Domshlak, C. 2009. Cost-optimal planning with landmarks. In *Proc. IJCAI 2009* (2009), 1728–1733.

Karpas, E., and Domshlak, C. 2011. Living on the edge: Safe search with unsafe heuristics. In *ICAPS 2011 Workshop on Heuristics for Domain-Independent Planning*.

Katz, M., and Domshlak, C. 2010a. Implicit abstraction heuristics. *JAIR* 39:51–126.

Katz, M., and Domshlak, C. 2010b. Optimal admissible composition of abstraction heuristics. *AIJ* 174(12–13):767–798.

Keyder, E.; Richter, S.; and Helmert, M. 2010. Sound and complete landmarks for and/or graphs. In *Proc. ECAI 2010*, 335–340.

Lipovetzky, N., and Geffner, H. 2011. Searching for plans with carefully designed probes. In *Proc. ICAPS 2011*, 154–161.

Mcallester, D., and Rosenblitt, D. 1991. Systematic nonlinear planning. In *Proc. AAAI 1991*, 634–639.

Nedunuri, S.; Cook, W. R.; and Smith, D. R. 2011. Cost-based learning for planning. In *ICAPS 2011 Workshop on Planning and Learning*, 68–75.

Pearl, J. 1984. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley.

Penberthy, J. S., and Weld, D. S. 1992. UCPOP: A sound, complete, partial order planner for ADL. In *Proc. KR 1992*, 103–114.

Pochter, N.; Zohar, A.; and Rosenschein, J. S. 2011. Exploiting problem symmetries in state-based planners. In *Proc. AAAI 2011*, 1004–1009.

Rintanen, J. 2003. Symmetry reduction for SAT representations of transition systems. In *Proc. ICAPS 2003*, 32–40.

Tate, A. 1977. Generating project networks. In *Proc. IJCAI 1977*, 888–893.

Vidal, V., and Geffner, H. 2006. Branching and pruning: An optimal temporal POCL planner based on constraint programming. *AIJ* 170(3):298–335.

Wolfe, J., and Russell, S. J. 2011. Bounded intention planning. In *Proc. IJCAI 2011*, 2039–2045.