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

# Partially Observable Online Contingent Planning Using Landmark Heuristics

**Shlomi Maliah**
Information Systems
Engineering
Ben-Gurion University
*shlomima@bgu.ac.il*

**Ronen I. Brafman**
Department of
Computer Science
Ben Ben-Gurion Universityv
*brafman@cs.bgu.ac.il*

**Erez Karpas**
CSAIL
MIT
*karpase@csail.mit.edu*

**Guy Shani**
Information Systems
Engineering
Ben-Gurion University
*shanigu@bgu.ac.il*

### Abstract

In contingent planning problems, agents have partial information about their state and use sensing actions to learn the value of some variables. When sensing and actuation are separated, plans for such problems can often be viewed as a tree of sensing actions, separated by conformant plans consisting of non-sensing actions that enable the execution of the next sensing action. This leads us to propose a heuristic, online method for contingent planning which focuses on identifying the next useful sensing action. The key part of our planner is a novel landmarks-based heuristic for selecting the next sensing action, together with a projection method that uses classical planning to solve the intermediate conformant planning problems. This allows our planner to operate without an explicit model of belief space or the use of existing translation techniques, both of which can require exponential space. The resulting Heuristic Contingent Planner (HCP) solves many more problems than state-of-the-art, translation-based online contingent planners, and in most cases much faster.

## 1 Introduction

Agents acting under partial observability acquire information about the true state of the world through sensing actions in order to achieve their goals. Planning for such problems is difficult, as one must consider multiple possible branches of execution that differ on the sensed values, resulting in potentially large plan trees. While one can generate a complete plan tree offline, a popular alternative is to replan after every observation, thus avoiding the computation of plans for cases that are not encountered within the current execution.

Existing contingent planners replan following an observation by computing a plan to achieve the goal under some assumptions on the value of future observations. For example, Albore, Palacios, and Geffner (2009) allow arbitrary values to be observed during execution. Thus, the underlying planner will typically make optimistic assumptions about their values in order to obtain a solution more quickly. An alternative approach is to make the observations correspond to some randomly chosen initial state (Shani and Brafman 2011) which is perhaps more realistic. Still, recent planners produce plans that attempt to achieve the *goal*, even though the plan is typically executed only until the next observation.



Our main contribution is a new online method for finding the most useful next sensing action, using a novel landmark-based heuristic. Once this sensing action is selected, we use a modified version of the planning domain to achieve the preconditions of that action. The process is repeated until the goal can be reached without additional information.

Our online Heuristic Contingent Planner (HCP) solves the original contingent problem using these ideas: repeatedly identifying a sensing action and solving the conformant planning problem of to achieving its preconditions. HCP is greedy, focusing on the next sensing action only. It is also approximate. Instead of using a conformant planner, it uses a classical planner over a projection of the conformant problem into a classical problem, circumventing the need for planning in belief-space.

Our approach builds strongly on the results of Bonet and Geffner (2011) for simple contingent planning domains. Bonet and Geffner show that simple domains can be solved using an efficient projection of the problem into classical planning. While their K-PLANNER plans directly over this translation, HCP uses this projection in two ways: First, it extends this projection (in a sound but incomplete manner) to handle non-simple domains, and uses it to derive landmarks for the original contingent problem that guide the selection of the next sensing action. Second, it uses this projection as a simple method for rapid conformant planning.

The sensing-action selection heuristic is the key novel part of HCP. The heuristic attempts to assess the "value-of-information" of alternative sensing actions by understanding their impact on the accumulated information. Specifically, the heuristic focuses on the number of landmarks achievable following each sensing action. We estimate this set of landmarks using landmarks of the classical projection of the original contingent problem. To improve the correspondence between the set of classical plans and the set of contingent plans, we add "inference" actions, that allow the classical planner to infer additional facts about the world following a sensing action. These facts are not explicitly described in the action's effects, but can be deduced from the effect and from information about the initial state.

We empirically evaluate HCP on a set of contingent planning benchmarks. Our experiments show that HCP is much faster than state-of-the-art contingent planners. It also creates shorter plans.

# 2 Contingent Planning

Contingent planning problems are characterized by uncertainty about the initial state of the world, partial observability, and the existence of sensing actions. Actions may be non-deterministic, but we, as well as most literature on contingent planning, will assume deterministic actions.

In this paper we focus on a restricted definition for contingent planning in which sensing is separated from actuation, i.e., an action either changes the world without yielding new information, or provides new information without changing the world (i.e., a sensing action). This is not a real restriction, as any planning domain can be transformed to satisfy the separation of sensing and actuation, and, indeed, all current benchmarks have this property.

## 2.1 Problem Definition

A contingent planning problem is a tuple: $\pi = \langle P, A_{act}, A_{sense}, \varphi_I, G\rangle$. $P$ is a set of propositions, $A_{act}$ is a set of actuation actions, and $A_{sense}$ is a set of sensing actions. $\varphi$ is propositional formula that describes the set of initially possible states. We assume that it is in *prime implicate form*. $G \subset P$ is the goal propositions.

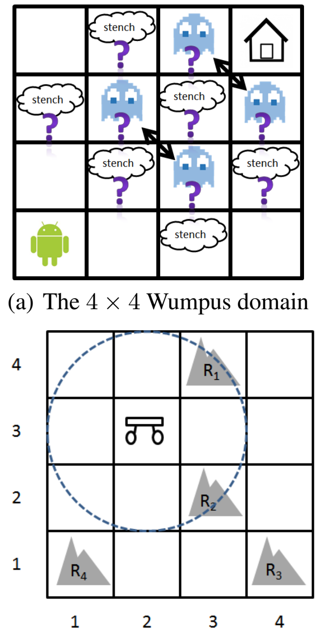
We often abuse notation, treating a set of literals as a conjunction of the literals in the set, as well as an assignment of the propositions in it. For example, $\{p, \neg q\}$ will also be treated as $p \wedge \neg q$ and as an assignment of *true* to $p$ and *false* to $q$. Given a literal $l$, we write $l \in P$ for a literal $l$ denoting that the (possibly negated) proposition in $l$ is in $P$. We use $\neg l$ to denote the negation of a literal where $\neg\neg p \equiv p$.

A state of the world, $s$, assigns truth values to all $p \in P$. A *belief-state* is a set of possible states. The initial belief state, $b_I = \{s : s \models \varphi_I\}$ is the set of initially possible states.

An actuation action $a \in A_{act}$ is a pair, $\{$*pre*$(a)$, *effects*$(a)\}$, where *pre*$(a)$ is a set of literal preconditions, and *effects*$(a)$ is a set of pairs $(c, e)$ denoting conditional effects. Following traditional classical planning conventions, all $p \in pre(a)$ must be positive. We use $a(s)$ to denote the state that is obtained when $a$ is executed in state $s$. If $s$ does not satisfy all literals in *pre*$(a)$, then $a(s)$ is undefined. Otherwise, $a(s)$ assigns to each proposition $p$ the same value as $s$, unless there exists a pair $(c, e) \in$ *effects*$(a)$ such that $s \models c$ and $e$ assigns $p$ a different value than $s$. We assume that $a$ is well defined, that is, if $(c, e) \in$ *effects*$(a)$ then $c \wedge$*pre*$(a)$ is consistent, and that if both $(c, e), (c', e') \in$ *effects*$(a)$ and $s \models c \wedge c'$ for some state $s$ then $e \wedge e'$ is consistent.

A sensing action $a \in A_{sense}$ is a pair, $\{$*pre*$(a)$, *obs*$(a)\}$, where *pre*$(a)$ is identical to the previous definition, and *obs*$(a)$ is a set of propositions in $P$ whose value is observed when $a$ is executed. That is, if $p \in$*obs*$(a)$ then following the execution of $a$, the agent will observe $p$ if $p$ holds, and otherwise it will observe $\neg p$. In contingent planning the agent must reason about the set of currently possible states, i.e. the belief state. Clearly, the observations affect the agent's belief state. We use $b_{a,o}$ to denote the belief state following the execution of $a$ in belief state $b$ and the observation $o$: $b_{a,o} = \{a(s) | s \in b, s \models o\}$. If $a$ is not applicable in all states in $b$ then $b_{a,o}$ is undefined.

A *conformant planning* problem is a contingent planning problem in which there are no sensing actions ($A_{sense} = \phi$).




(a) The $4 \times 4$ Wumpus domain

(b) The Rock Sample domain. The dotted line shows the range of the mineral sensor.

Figure 1: Two example domains.

We shall find it useful to separate $P$ into two sets: $P_k$ contains propositions whose value is always known, and $P_u = P \setminus P_k$ contains propositions whose value may be unknown (hidden) throughout the execution of (some parts of) a plan. These sets can be constructed recursively: $p \in P_u$ if its value is initially unknown as it does not appear in a unit-clause in $\varphi_I$. $p \in P_u$ also if there exists an action with conditional effect $(c, p)$ such that $c$ contains a literal from $P_u$. Given $P_k$ and $P_u$, we can separate $\varphi$ into two sets of clauses: $\varphi_k$ consists of unit clauses that assign the initial value of $P_k$, and $\varphi_u$ contains the remaining clauses, constraining the values of the propositions in $P_u$.

We assume deterministic and accurate observations. This assumption is restrictive if actions are deterministic, but in non-deterministic domains one can compile away non-deterministic observations. In any case, deterministic observations underlie all existing work in contingent planning.

We illustrate these ideas using a $4 \times 4$ Wumpus domain (Albore, Palacios, and Geffner 2009), which will serve as our running example. Figure 1(a) illustrates this domain, where an agent is located on a $4\times4$ grid. The agent can move in all four directions, and if moving into a wall, it remains in place. The agent initially is in the low-left corner and must reach the top-right corner. There are two monsters called Wumpuses hidden along the grid diagonal, the agent knows that Wumpus 1 can be at location 3,2 or 2,3, and Wumpus 2 can be at location 4,3 or 3,4. Thus the possible states are: $\{wat(3,2) \wedge wat(4,3), wat(3,2) \wedge wat(3,4), wat(2,3) \wedge wat(4,3), wat(2,3) \wedge wat(3,4)\}$. The stench of a Wumpus carries to all adjacent locations, and the agent can observe the stench in order to deduce the whereabouts of the Wumpuses.

In this problem $P_k$ contains the current position of the agent, and $P_u$ contains the possible stenches and Wumpus locations. Movement actions are in $A_{act}$ while smelling the stench of the Wumpus is in $A_{sense}$.

## 2.2 Solution Definition

Any plan for a contingent planning problem can be described as a tree $\tau = (N, E)$. Each node $n$ is labeled by an action $a(n)$ or by the distinguished action *noop*, denoting no operation to be taken by the agent. Each edge, $e$, is labeled with an observation $o(e)$. If $a(n)$ is labeled by a *noop*, it is a leaf node, denoting the plan termination. If $a(n) \in A_{act}$ is a non-sensing action, it has a single outgoing edge labeled by the null observation *true*. If $a(n) \in A_{sense}$ is a sensing action, it has one outgoing edge for each possible observation value, labeled by the corresponding observation.

We define $b(n)$ to be the belief state associated with node $n$ in the tree. For the root node, $b(n) = b_I$. Let $n'$ be a child of $n$ with ingoing edge $e$. Then $b(n') = b(n)_{a(n),o(e)}$.

We say that $\tau$ is a *complete solution plan* if for every leaf node $n$ in $\tau$, $b(n)$ is well defined and $b(n) \models G$. A *partial solution plan* is a plan tree where some outgoing edges, corresponding to possible observations, are missing.

## 2.3 Simple Contingent Problems

An interesting special case is the class of *simple* contingent planning problem (Bonet and Geffner 2011). These problems are characterized by two features. First, the value of any proposition that appears within an effect condition is initially known. This implies that $P_k$ consists of all propositions whose value is initially known, because if the value of a variable is initially known, it will always be known. Second, constraints on the value of initially unknown propositions are invariant. Thus, $\varphi_u$ is always true. An example of invariant hidden variables are the locations of the Wumpi above, which are initially unknown, never change, and do not affect the value of other variables.

# 3 The Fact Discovery Algorithm

We now formalize a generic, non-deterministic algorithm for online contingent planning, which we call the *fact discovery algorithm* (FDP). In each iteration, the algorithm checks if there is a conformant plan to the goal. If not, it non-deterministically chooses a sensing action *and* a conformant plan that achieves its preconditions, leading to an improved state of information. Note that there can be many different conformant plans that achieve an action's preconditions, and different choices impact our ability to reach the goal. The conformant planner returns *false* whenever no plan exists.

The online planner hence moves from one sensing action to another, until sufficient information has been gathered in order to reach the goal with no additional sensing actions. This is the main idea that underlies all online planners (Albore, Palacios, and Geffner 2009; Bonet and Geffner 2011; Shani and Brafman 2011; Brafman and Shani 2012a), but in this section we cast it in a more formal and general manner, showing that it is sound and complete.

**Algorithm 1** Fact Discovery Planner

**Input:** $\pi = \langle P, A_{act}, A_{sense}, \varphi_I, G\rangle$ — a contingent planning problem
1: $b \leftarrow b_0$ the initial belief state
2: **while** $b \not\models G$ **do**
3: $\quad P_{conformant} \leftarrow ChooseConformantPlan(P, A_{act}, b, G)$
4: $\quad$ **if** $P_{conformant} \neq false$: goal reachable w/o sensing **then**
5: $\quad\quad Execute(P_{conformant})$
6: $\quad$ **else**
7: $\quad\quad a_{obs} \leftarrow ChooseNextSensingAction(A_{sense}, b)$
8: $\quad\quad P_{conformant} \leftarrow ChoosetPlan(P, A_{act}, b, pre(a_{obs}))$
9: $\quad\quad$ **if** $P_{conformant}$ **= false then**
10: $\quad\quad\quad$ **return** *fail*
11: $\quad\quad$ **end if**
12: $\quad\quad Execute(P_{conformant}, b)$
13: $\quad\quad b \leftarrow \{P_{conformant}(s) | s \in b\}$: update belief state given $P_{conformant}$
14: $\quad\quad Execute(a_{obs})$ and observe literal $l$
15: $\quad\quad b \leftarrow b_{a_{obs},l}$ : update the belief state given $l$
16: $\quad$ **end if**
17: **end while**

**Claim 1.** *Algorithm 1 is a sound and complete contingent planning algorithm if the underlying conformant planner is sound and complete.*

*Proof. Completeness:* Given a solution contingent plan $\tau$, every path in $\tau$ from the root to a leaf is a sequence of actions $\langle a_1, ..., a_n\rangle$, containing $k \in [0, n]$ sensing actions. Proof is by induction on $k$. If $k = 0$ then $\langle a_1, ..., a_n\rangle$ is a conformant plan that achieves the goal, which can be chosen in step 3 of the algorithm. Otherwise, when $k > 0$, let $a_i$ be the first sensing action, where $i \in [1, n]$. FDP can choose $a_i$ as its first $a_{obs}$ in step 7. Then, $a_0, ..., a_i$ can be chosen in step 8 for achieving the preconditions of $a_i$. After executing $a_i$, we remain with a sequence containing $k-1$ sensing actions, and hence, by induction, FDP can continue making appropriate choices to complete the sequence and achieve the goal.

*Soundness:* As the conformant planner is sound, it will always return a valid conformant plan achieving the preconditions of $a_{obs}$. Thus, $a_{obs}$ can be soundly executed immediately after the conformant plan. Hence, all the actions have well-defined outcomes, and achieve the intended goal. □

Of course, non-deterministic planning is of theoretical interest only. Furthermore, iteratively solving a conformant planning problem is costly. An effective implementation of FDP must smartly and efficiently choose the next sensing action and conformant plan. Section 4 will consider methods for replacing conformant planning with classical planning. Section 5 will describe a method for replacing the non-deterministic choice in line 7 with greedy choice informed by a landmarks-based heuristic.

However, before we consider efficient implementations, we present another theoretical result, based on ideas appearing in (Bonet and Geffner 2011). We show that in dead-end free domains, using a greedy choice results in a sound and complete planning algorithm.

We say that a contingent planning problem $\pi = \langle P, A_{act}, A_{sense}, \varphi_I, G\rangle$ is *dead-end free* if $\langle P, A_{act}, A_{sense}, b, G\rangle$ is solvable for any belief state

$b$ reachable from $\varphi_I$ in $\pi$. Define *FDP-Progress* to be a deterministic implementation of FDP such that the sensing action $a_{obs}$ selected in lines 7 senses a proposition whose value is unknown prior to the execution of $a_{obs}$, whenever this is possible. Now, we can prove the following result.

**Claim 2.** *In dead-end free domains, FDP-Progress will lead to the goal in a finite number of steps.*

*Proof.* First, suppose that in some belief state it is not possible to learn new information. That is, there is no conformant plan that achieves the preconditions of any informative sensing action. In that case, since the domain is dead-end free, there must be a conformant plan that reaches the goal. Otherwise, each call to steps 7 and 8 yields a sensing action, revealing the value of some unknown proposition $l$ which reduces the size of the belief state. That is, some states in $b$ must disagree with the observed $l$ in line 14 (otherwise $l$ is known prior to the execution of $a_{obs}$) and these states are removed from $b$, reducing its size. After at most $|b| - 1$ iterations, if the goal is not reached beforehand, we reach a belief state of size 1, in which case the problem is reduced to a classical setting, which is solvable as there are no dead-ends. Notice that this proof does not hold when actions have non-deterministic effects, as the belief state size can grow. □

In some domains, identifying an action that reduces the size of the belief state may require non-trivial effort because $a_{obs}$ must provide information with respect to the belief state where it is executed, following the conformant plan, rather than the belief state where it is chosen. In other domains (e.g., simple domains) it is easy to identify such actions.

Our actual planner is essentially FDP-Progress with an efficient heuristic for sensing-action selection and an efficient (incomplete) conformant planner that uses a classical approximation, which we describe below. As shown in our experiments, this combination leads to a very efficient and effective online contingent planner.

# 4 Classical Projections for Conformant and Contingent Planning Problems

Contingent planning requires maintaining and manipulating an implicit or explicit description of the agent's belief state, which can be exponential in size. The same holds for conformant planning, which can be viewed as a special case of contingent planning in which there are no sensing actions. In this section we describe projection techniques for reducing contingent and conformant planning problems into classical planning that are based on a transformation introduced in (Bonet and Geffner 2011) for simple contingent planning domains. Sometimes, these methods are sound and complete — specifically, in simple domains. In non-simple domains they provide a sound approximation. Our algorithm uses these techniques in two ways. First, to replace conformant planning with classical planning. Second, to allow the use of classical landmark detection algorithms (Hoffmann, Porteous, and Sebastia 2004) to detect landmarks in contingent planning problems. These landmarks play a key role in our heuristic for selecting the next sensing action (line 7 in the algorithm). When defining projections, we will initially make various restrictions (e.g., positive preconditions) to make the definitions simpler to follow, which will gradually be removed.

## 4.1 Conformant Planning Approximations

**Definition 1.** *Let* $\pi = \langle P, A_{act}, \varphi, G\rangle$ *be a conformant planning problem. The* classical projection *of* $\pi$ *is a classical problem* $\langle P, A_{act}, I, G\rangle$ *where:* $I = \varphi_k \wedge \bigwedge_{p \in P_u} \neg p$.

That is, propositions in $P_u$ are all assigned *false*, initially, while those in $P_k$ are initialized as in $\pi$. Intuitively, the weak projection makes a pessimistic assumption concerning the hidden variables, assuming all their values to be initially false. This assumption is pessimistic because we assume positive (pre)conditions only, so with negative values, we restrict our ability to apply actions.

**Lemma 1 (Bonet&Geffner 2011).** *$\rho$ is a conformant plan for a simple conformant planning problem iff it is a classical plan for its classical projection.*

We can now define *FDP-classical-project* to be a variant of FDP in which conformant planning is performed over the classical projection of the problem and conclude:

**Corollary 1.** *Under the corresponding assumptions on the classical planner, FDP-classical-project is a sound and complete online contingent planner for simple contingent planning problems.*

## 4.2 Contingent Planning Approximations

Moving from conformant to contingent problems, we must extend the definition of the classical projection to handle observations. Intuitively, a projection not only maps a contingent planning domain into a classical planning domain, but also associates a branch in a (possibly partial) solution tree with a (classical) plan generated for the projection. This correspondence is typically straightforward (e.g., a one-to-one mapping of actions), so we will not define it formally.

We start with the special case of simple contingent domains (Bonet and Geffner 2011). We use a three-valued representation of each proposition, denoting three possibilities: $p$ is true in all possible states, $p$ is false in all possible states, and $p$ has different values in different currently possible states. This provides a rough approximation of the belief state of the agent, which in simple contingent domains is sufficient. To simplify the notation, we treat $p$ as a 3-valued variable, $p^*$, with possible values $\{t, f, u\}$, which is a syntactic sugar for the of use two variables: $Kp$ and $K\neg p$, where $p^* = t$ denotes $Kp \wedge \neg K\neg p$, $p^* = f$ denotes $K\neg p \wedge \neg Kp$, and $p^* = u$ denotes $\neg Kp \wedge \neg K\neg p$. We will abuse notation writing $l^* = t, f, u$ implying $p^* = t, f, u$ if $l = p$ and $p^* = f, t, u$ for $l = \neg p$. Finally, for $p \in P_k$, $p = u$ never holds, so we can treat it as a standard proposition, but for the sake of uniformity, we will not make this distinction.

### The Simple Weak Projection

**Definition 2.** *Let* $\pi = \langle P, A_{act}, A_{sense}, \varphi, G\rangle$ *be a simple contingent planning problem. The* simple weak contingent projection *of* $\pi$ *is a classical problem* $\langle P^*, A'_{act} \cup A'_{sense}, X, \varphi_k \wedge \bigwedge_{p \in P_u} p = u, G^*\rangle$*, where:*

- $P^* = \{p^* | p \in P\}$

- $A'_{act}$ *contains an action* $a'$ *for every* $a \in A_{act}$*, replacing every* $p$ *in* $a$ *by* $p^* = t$ *in* $a'$ *and every* $\neg p$ *with* $p^* = f$.
- $G^* = \{g^* = t | g \in G\}$.
- *For each action* $a \in A_{sense}$ *and every observable proposition* $p \in obs(a)$ *there are two corresponding actions* $a_p^+$ *and* $a_p^-$ *in* $A'_{sense}$*. Both* $a_p^+$ *and* $a_p^-$ *have the same preconditions as* $a$*, except that* $l$ *is replaced by* $l^* = t$*, as well as one additional precondition* $p^* = u$*. The effect of* $a_p^+$ *is* $p^* = t$*, and the effect of* $a_p^-$ *is* $p^* = f$.
- $X$ *is a set of axioms (Thiébaux, Hoffmann, and Nebel 2005). For each* invariant *clause* $c \in \varphi_u$ *and each literal* $l$ *in* $c$ *we create an axiom* $x_{c,l} \in X$*. The axiom's preconditions are* $\bigwedge_{l' \in c, l' \neq l} l'^* = f$ *and its effect is* $l^* = t$.
- $I$ *assigns* $t$ *to* $p^*$ *if* $\varphi \models p$*,* $f$ *if* $\varphi \models \neg p$*, and* $u$*, otherwise.*

The axioms provide a method for reasoning about the value of propositions given observations and the invariant statements in $\varphi_u$. While axioms can be implemented as simple actions, the benefit of using axioms is that we get a cleaner correspondence between classical plans and branches of (possibly partial) contingent plans. Recall that in simple domains, all clauses in $\varphi_u$ are invariant.

**The Weak Projection** To ensure that the weak projection is sound in general (possibly non-simple) contingent domains, we need to enhance the projection by allowing known variables to become unknown. We assume here that effect conditions are conjunctions of literals (straightforward to generalize). Let $c = l_1 \wedge \cdots \wedge l_k$ be an effect condition. We define $c = u \equiv \bigvee_i (l_i = u)$. Intuitively, $c = u$ holds if the value of $c$ is not the same in all possible states.

**Definition 3.** *The (general) weak projection is identical to the simple weak projection, except that the actions are modified as follows:*

- *For every conditional effect* $(c, l)$*, add a new conditional effect* $(c = u \wedge l* = f, l* = u)$.

When the contingent domain is simple, the weak projection is equivalent to the simple weak projection, which is the encoding used by K-PLANNER to handle simple domains. However, the (general) weak projection can be used in any contingent planning domain.

Beyond our generalization of the projection method, which we use to handle arbitrary contingent domains, there is an important difference in how the projections are used by K-PLANNER and our planner. K-PLANNER uses the projection in the contingent case in order to generate a plan for the contingent problem. We generate plans only for the intermediate, shorter and simpler *conformant problems*, using the classical projection, which is simpler than the weak projection of the general contingent problem. We use the weak projection of the contingent problem only as a tool for approximating the set of landmarks of the contingent problem. This set of landmarks is used to inform our heuristic choice of the next sensing action, as described in the next section. Moreover, unlike K-PLANNER , the projection of sensing actions is not used to make optimistic assumptions concerning the path to the goal by the planner, but rather to capture all possible paths to the goal, and our heuristic will weigh all possible outcomes of a sensing action.

## 4.3 Soundness and Completeness

The weak projection provides a general technique for generating a classical approximation of a contingent planning problem. Unfortunately, the correspondence between classical solutions of the projection and plan branches in the contingent solution, is not straight forward, even though each projection action corresponds to a contingent action. This is because, in some cases, the observations selected in the projection plan may not correspond to the execution of the corresponding contingent branch in any possible initial state. On the other hand, there can be branches in the contingent tree that do not correspond to any valid plan of the projection. Still, we can formalize a correspondence between projection plans and contingent branches.

There is a natural "belief-state" associated with each classical state of the projected problem — given a classical state $s$ over the variables $P^*$, $b_c(s)$ will be used to denote the set of states over the original propositions in $P$ that agree with $s$ on all variables assigned $t$ and $f$. I.e., if $p^* = t$, $b_c(s)$ contains only states in which $p$ holds, and if $p^* = f$, $b_c(s)$ contains only states in which $\neg p$ holds. Otherwise, when $p^* = u$, there are no constraints on the value of $p$.

Let $b_{\bar{a}\bar{o}}$ be the belief state in the contingent problem domain $\pi$, following the execution of a sequence $\bar{a}\bar{o}$ of actions and observations. Let $s_{\bar{a}\bar{o}}$ be the classical state in some classical projection $\pi_c$. We say that the projection is *sound* if for every sequence $\bar{a}\bar{o}$, we have that $b_{\bar{a}\bar{o}} \subseteq b_c(s_{\bar{a}\bar{o}})$.

Interestingly, we do not necessarily need to generate an interpretation that satisfies $b(s) = b$ to faithfully model a problem. A classical projection is *complete* when $b(s) \models l$ iff $b \models l$, for every literal $l$ that appears in an action's precondition or in $G$. Completeness does not imply that the classical projection maintains the same state, but that it maintains enough information to ensure that the plan it generates is indeed executable, and that it achieves the goal.

Although their definitions are somewhat different than ours, the theorems of (Bonet and Geffner 2011) imply:

**Lemma 2 (BG11).** *Weak projections are sound and complete for simple contingent planning domains.*

For the general case we can show:

**Lemma 3.** *The (general) weak projection is sound.*

The proof is by induction on the length of the action sequence An obvious implication of soundness is that when a plan for a sound projection achieves the goal, and the observations along this plan are consistent with some initial state $s$, then the execution of the corresponding plan branch when $s$ is the true initial state will also achieve the goal.

## 4.4 Stronger Projections

The weak projection is, in general, incomplete. That is, many valid branches of contingent plans do not correspond to a valid plan of the projected problem. Some methods can generate a sound and complete approximation for general contingent planning domains — specifically, the transformations used by CLG (Albore, Palacios, and Geffner 2009) and SDR (Brafman and Shani 2012b). Unfortunately, these methods generate exponentially larger classical problems.

**Algorithm 2** Heuristic Contingent Planner

**Input:** $\pi$ — a contingent planning problem

$\pi_{enhanced} \leftarrow$ the enhanced weak projection of $\pi$
$L \leftarrow$ the set of landmarks for $\pi_{enhanced}$
$s \leftarrow$ all known literals at the initial state
**while** there is no solution to the classical projection of $\pi$ **do**
  $a_{obs} \leftarrow ChooseNextSensingAction(\pi, L, s)$
  $P \leftarrow$ a conformant plan for $\pi_{classical}$, with $G = pre(a_{obs})$
  $s' \leftarrow Execute(P, s)$
  $Execute(a_{obs})$ and observe literal $l$
  $s \leftarrow s' \cup \{l\}$
**end while**
$P \leftarrow$ a conformant plan for the classical projection of $\pi$
$Execute(P, s)$

It is thus natural to seek stronger projections, in the sense that more branches of contingent plans correspond to plans of the projection, while maintaining an economical projection size, preferably linear. A projection becomes stronger as it reduces the difference between $b(s)$ and $b$, that is, it allows us to rule out additional states that are in $b(s)$, but not in $b$. One can generate stronger sound approximations by, e.g., adding additional sound inference axioms to the projection.

Formally, let $\pi_{c_1}$ and $\pi_{c_2}$ be two sound classical projections of $\pi$. We say that $\pi_{c_1}$ is *at least as strong as* $\pi_{c_2}$ if $b_{c_1}(s_{\bar{a}\bar{o}}) \subseteq b_{c_2}(s_{\bar{a}\bar{o}})$.

The obvious benefit of strengthening the approximation is that more problems become solvable. That is, stronger approximations may be able to generate plans that correspond to observation sequences that the weaker approximation cannot handle, and will always be able to handle observation sequences that the weaker approximation handles.

We now suggest one class of projected actions that can be used to strengthen the weak approximation; in some benchmarks the observed values conditionally depend on a previously executed action. Learning about the observed value can then help us deduce knowledge over the effect condition. We illustrate this idea using the RockSample domain (Smith and Simmons 2004; Brafman and Shani 2012b).

In this domain, the rover must sample rocks containing a desirable mineral in a grid (Figure 1(b)). The rocks locations are known, but the agent must sense for the mineral using a long range sensor. The sensor reports the existence of the mineral in some rock within its range.

A natural formalization of this domain may have an action *activate-sensor-at-2-3* with preconditions *at-2-3* and conditional effects *good-rock*$_1 \rightarrow$ *good-rocks-in-range* and *good-rock*$_2 \rightarrow$ *good-rocks-in-range*, and an additional action *observe-rocks-in-range* which observes the *good-rocks-in-range* proposition which signifies the sensor's output. While these are separate actions, used together they allow us to reason about which rocks contain the good mineral. In this case, *at-x-y* is always known, *good-rocks-in-range* is unknown but directly observable, and *good-rock*$_i$ is unknown and not directly observable. To address this, we generate a new action that corresponds to the sequential activation of these two actions — *activate-sensor* and *observe-good-rocks-in-range* — with effects reflecting the knowledge obtained by the latter observation.

**Definition 4.** *Let $a$ be an action with conditional effects*

**Algorithm 3** Choosing the Next Sensing Action

**Input:** $\pi_{classical}$ — a classical projection, $L$ a set of landmarks, $s$ the set of currently known literals

$s' \leftarrow$ the set of achievable literals using $\pi_{classical}$ starting from $s$ (approximated using delete-relaxation)
$\Omega \leftarrow \{a : a \in A_{sense}, pre(a) \in s', obs(a) \notin s'\}$
**for each** action $a \in \Omega$ **do**
  $p \leftarrow obs(a)$, $s'_+ \leftarrow s' \cup \{p\}$, $s'_- \leftarrow s' \cup \{\neg p\}$
  $s''_+ \leftarrow$ the set of possible literals given $\pi_{classical}$ and $s'_+$
  $s''_- \leftarrow$ the set of possible literals given $\pi_{classical}$ and $s'_-$
  $h_{lm}(a) \leftarrow$ # possible landmarks in $s''_+$ and $s''_-$, but not in $s'$
  $h_{li}(a) \leftarrow$ # possible literals in $s''_+$ and $s''_-$, but not in $s'$
  $h_{obs}(a) \leftarrow$ # possible sensing actions in $s''_+$ and $s''_-$, but not in $s'$
  $h_{cost}(a) \leftarrow$ # required actions from $s$ before $a$ can be executed in $\pi_{classical}$
  $h(a) \leftarrow \langle h_{lm}(a), h_{li}(a), h_{obs}(a), h_{cost}(a)\rangle$
**end for**
**return** $argmax_{a \in \Omega} h(a)$

*$(c, p)$ and $(\neg c, \neg p)$ where $c$ is an unknown and unobservable literal and $p$ is observable through $a_{obs}$, and $a$ does not negate any precondition of $a_{obs}$ and does not change the value of $c$. Define the classical action $a \circ a^+_{obs}$ as follows: $pre(a \circ a^+_{obs}) = pre(a) \cup pre(a_{obs})$, and $effects(a \circ a^+_{obs}) = effects(a) \wedge p \wedge c$, where $effects_u(a)$ are the unconditional effects of $a$, replacing in the process every occurrence of a literal $l$ with $l^* = t$ and every $\neg l$ with $l^* = f$ (as in the definition of the weak projection). Similarly, define $pre(a \circ a^-_{obs}) = pre(a) \cup pre(a_{obs})$, and $effects(a \circ a^-_{obs}) = effects(a) \wedge \neg p \wedge \neg c$, again with literals in $P_u$ treated as above. The* enhanced weak projection *of a contingent problem is its weak projection, enhanced with these actions.*

*More generally, when $c = \bigvee c_i$ is a disjunction of literals, define $a_i \circ a^+_{obs}$ as above, except that the preconditions of $a \circ a_{obs}$ contain $\bigwedge_{j \neq i} \neg c_j$ as well. The effect is now $c_i$, rather than $c$. And similarly for $a_i \circ a^-_{obs}$*

In the small RockSample example in Figure 1(b) we have two such joined actions with preconditions *at-2-3* $\wedge\, \neg$*good-rock*$_j$ and effect *good-rock*$_i$ where $i = 1, j = 2$ or $i = 2, j = 1$ (the observation *good-rocks-in-range* is removed in the classical projection).

For clarity of the correspondence definition, note that the additional actions $a \circ a^{+/-}_{obs}$ are similar to $a^+_p$ and $a^-_p$.

**Lemma 4.** *The enhanced weak projection is a sound approximation which is at least as strong as the weak projection.*

The idea behind this enhancement can be farther generalized as follows:[1] Assume a sequence of unobservable propositions $p_1, ..., p_k$ and a set of actions $a_i$ with conditional effects $(p_i, p_{i+1})$, and an observation action that applies only to $p_k$. Having observed $p_k$, we can deduce that all $p_i$ hold, whereas the above inference action allows deducing $p_{k-1}$ only. We note that the *localize* domain exhibits such behavior, while many others (e.g. *medpks, rock-sample*) require only actions of the form $a \circ a_{obs}$ as above.

[1] This generalization was not implemented.

Table 1: Comparing the performance of contingent planners. We report only the largest instances that were solved of each benchmark. Blank cells represent problems that the planners were unable to solve. CSU denotes models that CLG can solve but cannot simulate execution for. PF denotes planner failure. N/A denotes that the planner is inapplicable for that domain. Results are averaged over 50 executions, and we report standard error in parenthesis.

| | HCP | | | MPSR | | SDR | | CLG | | K-PLANNER | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Landmarks | Actions | Time | Actions | Time | Actions | Time | Actions | Time | Actions | Time |
| cloghuge | 9 | 53.55 (0.11) | **1.8** (0.019) | PF | | 61.17 (0.44) | 117.13 (4.19) | **51.76** (0.33) | 8.25 (0.08) | N/A | |
| ebtcs-70 | 3 | **34.5** (0.17) | **0.36** (0.015) | 44.5 (0.7) | 22.4 (0.3) | 35.52 (0.75) | 3.18 (0.07) | 36.52 (0.86) | 73.96 (0.14) | N/A | |
| elog7 | 5 | **19.9** (0.056) | **0.084** (0.004) | 23.5 (0.1) | 1.4 (0.1) | 21.76 (0.07) | 0.85 (0.01) | **20.12** (0.05) | 1.4 (0.08) | N/A | |
| CB-9-5 | 29 | **320** (0.31) | **57.7** (0.132) | PF | | 392.16 (2.81) | 505.48 (8.82) | CSU | | 358.08 (15.8) | 94.18 (3.31) |
| CB-9-7 | 37 | **425** (0.243) | 161.5 (0.137) | PF | | 487.04 (2.95) | 833.52 (15.82) | CSU | | 458.36 (14.64) | **116.63** (3.24) |
| doors15 | 16 | **143** (0.201) | **17.7** (0.0.061) | 262.2 (1.9) | 190 (3.3) | 143.24 (1.36) | 268.16 (3.78) | PF | | 150.88 (4.7) | **55.24** (2) |
| doors17 | 18 | **184** (0.23) | **46.11** (0.205) | 368.25 (3.4) | 335.3 (5.3) | 188 (1.64) | 416.88 (6.16) | PF | | 188.8 (5.79) | 79.24 (2.62) |
| localize17 | | N/A | | 59.8 (0.9) | **230.4** (7.7) | **45** (0.86) | 928.56 (33.2) | CSU | | N/A | |
| unix3 | 1 | **42.0** (0.19) | **0.67** (0.021) | 69.7 (1.7) | 5.2 (0.1) | 56.32 (1.72) | 5.47 (0.18) | 51.32 (0.97) | 18.56 (0.05) | **45.48** (4.59) | 16.87 (1.56) |
| unix4 | 1 | **76.55** (0.286) | **7.25** (0.074) | 158.6 (4.3) | 30.4 (1.1) | 151.72 (4.12) | 35.22 (0.94) | 90.8 (2.12) | 189.41 (0.6) | 87.04 (8.54) | 38.81 (3.53) |
| Wumpus15 | 36 | **65.08** (0.223) | **2.33** (0.049) | **65** (1.6) | 126.6 (3.1) | 120.14 (2.4) | 324.32 (7.14) | 101.12 (0.67) | 330.54 (0.25) | 107.64 (4.6) | 7.17 (0.6) |
| Wumpus20 | 46 | 90 (0.273) | **5.16** (0.075) | **71.6** (1.2) | 261.1 (7) | 173.21 (3.4) | 773.01 (20.78) | 155.32 (0.95) | 1432 (0.47) | 151.52 (6.29) | 16.03 (1) |
| RockSample 8-12 | 20 | **115** (0.135) | **0.50** (0.005) | PF | | 127.24 (0.68) | 113.4 (0.79) | N/A | | N/A | |
| RockSample 8-14 | 22 | **135** (0.144) | **0.65** (0.015) | PF | | 142.08 (0.8) | 146.75 (1.19) | N/A | | N/A | |

### 4.5 Approximate Landmarks through Projections

We have seen the correspondence between plans for sound projections and branches of plan tree for contingent problems. This motivates a simple and useful technique for generating an approximate set of landmarks for the contingent planning by computing the set of landmarks of a classical projection. The advantage of this reduction is that, again, we can employ off-the-shelf techniques for classical landmark generation to generate an approximate set of landmarks for the contingent problem. This set will be used to guide the selection of the next sensing action.

In simple domains, as the weak projection is sound and complete, the landmarks detected over the projection are the true landmarks of the contingent problem. In non-simple domains, however, there is no clear relationship between landmarks of the classical projection and landmarks of the original contingent problem. The reason is that some classical solutions may not correspond to any branch of any solution plan, and that some branches of a solution plan may not correspond to classical solutions. Still, the landmarks of the classical projection are a strong heuristic for the contingent problem, as our experiments demonstrate.

## 5 The Heuristic Contingent Planner

We now present our online contingent planner (Algorithm 2). The planner progresses by repeatedly identifying a reachable sensing action that heuristically provides valuable information towards the achievement of the goal. The planner then plans in the classical projection for the achieving the preconditions of the observation action. The plan is executed, followed by the observation action. Now, the process repeats with the additional information that was provided by the observation action. This process is repeated until the goal can be achieved without any additional sensing actions.

Algorithm 3 shows the process for selecting the next sensing action. It estimates the myopic value of information of the observation action, i.e., how much value will be achieved from executing the action, ignoring future observations. This value is estimated using the number of disjunctive action landmarks that can be achieved following the sensing actions. The set of landmarks computed is the set of landmarks for the enhanced weak projection.

First, we compute the set of achievable literals in the classical projection, that is, without any sensing actions. Then, we see which observation actions that sense the value of some unknown proposition can be executed given the set of achievable literals. These are the candidate actions to be returned by the algorithm. To choose the heuristically best observation action, we analyze the value of observing $p$, by assuming once that we have observed $p$, and once $\neg p$ and computing which literals become reachable in each case.

Our policy for returning the heuristically best action first looks at the number of satisfiable landmarks following the observation. Given multiple observation actions that satisfy the same number of landmarks, we break ties by looking at the sum of the number of literals and new observation actions that become achievable following the execution of the observation action. Finally, we break ties in favor of the action which we estimate (using delete-relaxation) to require the minimal number of actions to execute.

As we explained above, the planner is sound and complete for simple domains, but can be used in non-simple domains. In non-simple domains the classical projection we use to compute a conformant plan to the next sensing action can be incomplete. We did not run across this problem in our experiments, and theoretically, one can call a true conformant planner if the classical projection fails to plan for a sensing action's preconditions or the goal.

As the classical projection is sound, the returned plan is also sound, and achieves the preconditions of the sensing action. On the other hand, it may well be that the execution of the sensing action will not provide the intended new information. This is because the choice of the conformant plan may be critical for obtaining new information, and the classical projection may fail to identify such useful plans. In these cases, our algorithm may be unable to discover new information, rendering it incomplete yet sound.

## 6 Empirical Evaluation

We compare HCP to state-of-the-art online contingent planners, CLG (Albore, Palacios, and Geffner 2009), SDR (Brafman and Shani 2012b), MPSR (Brafman and Shani 2012a), and K-PLANNER (Bonet and Geffner 2011). The experiments were conducted on a Windows Server 2008 machine with 2.66GHz cores and 32GB of RAM. The underlying classical planner is FF (Hoffmann and Nebel 2001) — translated problems are written to PDDL and then FF is run in a different process. To test the soundness of action preconditions, and whether the goal was obtained, we use regression (Shani and Brafman 2011).

Table 1 shows the runtime of the planners (including the time required to identify landmarks for HCP), showing HCP to be much faster than all other planners on almost all problems. The plan quality (number of actions) of HCP is also good. The only domain that could not be solved by HCP is *localize*, because it does not conform to our assumptions concerning reasoning about the hidden propositions. We also report the number of landmarks discovered by HCP in each problem. As can be seen, even in domains like Unix, which have no useful landmarks, HCP still works very well due to the other heuristic components.

As Algorithm 3 has several heuristic components, we analyze the components separately in Table 2. C denotes the minimal cost heuristic (Line 10), P denotes the number of literals heuristic (Line 8), A denotes the number of sensing actions heuristic (Line 9), and L denotes the number of landmarks heuristic (Line 7). In this problem, using the sensing action with the minimal cost (C) performs the worst, because it always selects the nearest sensing which is not always very useful. Looking at the number of additional available sensing actions (A) is also not useful here. The number of new achievable literals following the sensing action (P) and the number of new landmarks (L) both result in relatively shorter plans, where the number of literals is somewhat better. Looking at the total time, though, we can see that the landmarks approach works much faster. This is because the landmarks heuristic uses much less sensing actions, as the selection of each sensing action requires the relatively costly heuristic estimation in Algorithm 3, then less sensing actions result in faster computation. Furthermore, we can see that the combination of the various heuristics greatly improve on the overall runtime, by combining the strengths of the various components.

Table 2: Evaluating heuristic components for Wumpus 15. We show the average number of actions and sensing actions to the goal, and the time for completion.

| Heuristic | Actions | Sensing | Time |
|---|---|---|---|
| C | 225 | 22 | 57.3 |
| P | 138 | 25 | 13.5 |
| A | 349 | 22 | 11.1 |
| L | 162 | 7 | 3.43 |
| L,C | 81 | 6 | 3.09 |
| L,C,P | 65 | 5 | 2.33 |
| All | 65 | 5 | 2.33 |

## 7 Discussion and Related Work

Bonet and Geffner (2000) were the first to propose using heuristic search in belief space. Since then, several heuristics for belief states were proposed. Bryce, Kambhampati, and Smith (2006) argue that belief space heuristics typically aggregate distance estimates from the individual states within the belief state to the goal. For example, the DNF planner (To, Pontelli, and Son 2009) uses a heuristic based on the number of satisfied goals, the cardinality of the belief state, and the sum of the number of unsatisfied goals in each individual state in the belief state.

HCP is related to these planners as it also searches in belief space, although it doesn't explicitly represent and reasons about it. Our use of landmark detection is more useful than existing heuristics of other belief search planners, and it is hard to see it as aggregating heuristic estimates of individual states. HCP is also one of the first belief-space search planners to use online planning – indeed, the focus on offline search by previous planners in this class reduced their scalability.

A second popular approach to contingent planning is the compilation-based approach (Albore, Palacios, and Geffner 2009; Shani and Brafman 2011; Brafman and Shani 2012a; Bonet and Geffner 2011) in which a contingent problem is translated into a classical planning problem. This method allows leveraging advances in classical planning, such as recent, powerful heuristic generation methods. Online methods employing compilation are popular, planning only for reachable branches of the plan tree given the true hidden state at runtime. HCP uses translation techniques to solve the intermediate conformant planning problems.

We leverage both the theoretical and the practical ideas presented by Bonet and Geffner (2011). First, we use their compilation techniques to apply classical landmark detection to contingent problems, through the K-PLANNER translation. Furthermore, instead of using a sound and complete, and therefore slower, conformant planner, we plan in the classical projection, which was presented by K-PLANNER , and extended here to some non-simple problems. In a way, at least for simple problems, our approach can be considered as introducing a landmark heuristic into K-PLANNER . An important part of K-PLANNER is the use of the projection to provide an efficient belief maintenance technique. This technique, however, is incomplete for non-simple domains, losing much information. We do not maintain an explicit belief-state, but instead use regression to verify properties of the current belief state required for planning.

In classical planning, landmarks were initially used to identify subgoals. Later on, researchers achieved better results by using landmarks as a heuristic for guiding the search (Richter and Westphal 2010). In this paper we return to the idea of using landmarks to identify subgoals, which are in our case observation actions. We speculate that our online replanning approach, where new information is discovered throughout the execution, is much more suitable for subgoaling than classical planning, where the results of applying an action are always known offline. In our replanning approach, where sensing actions are separated by conformant plans, treating the next sensing action as a subgoal greatly reduces the search space in the conformant space, which is exponentially larger than that of the classical underlying problem.

In Section 4.4 we suggest how some actions can be composed to create a new projected action. There is ongoing interest in planning research in identifying macro actions, composed of a number of atomic actions (Newton 2009). Our method is however much simpler, focusing on a popular specific action composition, as opposed to more general methods that automatically identify sequences of actions to be composed into macro actions. It is possible that smarter identification of macro actions can be further used to allow us to solve additional contingent domains.

## 8 Conclusion and Future Work

We introduced a new approach to contingent planning, relying on heuristics computed over a projection of the domain description, without maintaining a belief state explicitly, or translating the problem into classical planning, which are the two popular approaches to contingent planning under partial observability.

Our planner, HCP, leverages properties of many benchmarks in order to compute landmarks. Domains which do not conform to these properties (e.g. Localize), cannot be solved by HCP. In the future we will create stronger projections, allowing us to solve additional domains.

## Acknowledgments

This research is partially supported by ISF grant 933/13, the Paul Ivanier Center for Robotics Research and Production Management, and the Lynn and William Frankel Center for Computer Science. Erez Karpas is partially supported by the DARPA MRC Program, under grant number FA8650-11-C-7192, and Boeing Corporation, under grant number MIT-BA-GTA-1.

## References

Albore, A.; Palacios, H.; and Geffner, H. 2009. A translation-based approach to contingent planning. In *IJCAI*, 1623–1628.

Bonet, B., and Geffner, H. 2000. Planning with incomplete information as heuristic search in belief space. In *Proc. AIPS'00*, 52–61.

Bonet, B., and Geffner, H. 2011. Planning under partial observability by classical replanning: Theory and experiments. In *IJCAI*, 1936–1941.

Brafman, R. I., and Shani, G. 2012a. A multi-path compilation approach to contingent planning. In *AAAI*.

Brafman, R. I., and Shani, G. 2012b. Replanning in domains with partial information and sensing actions. *Journal of Artificial Intelligence Research (JAIR)* 45:565–600.

Bryce, D.; Kambhampati, S.; and Smith, D. E. 2006. Planning graph heuristics for belief space search. *JOURNAL OF AI RESEARCH* 26:35–99.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *JAIR* 14:253–302.

Hoffmann, J.; Porteous, J.; and Sebastia, L. 2004. Ordered landmarks in planning. *J. Artif. Int. Res.* 22(1):215–278.

Newton, M. A. H. 2009. *Wizard: Learning Macro-Actions Comprehensively for Planning*. Ph.D. Dissertation, University of Strathclyde.

Richter, S., and Westphal, M. 2010. The lama planner: Guiding cost-based anytime planning with landmarks. *J. Artif. Int. Res.* 39(1):127–177.

Shani, G., and Brafman, R. I. 2011. Replanning in domains with partial information and sensing actions. In *IJCAI*, 2021–2026.

Smith, T., and Simmons, R. 2004. Heuristic search value iteration for POMDPs. In *UAI 2004*.

Thiébaux, S.; Hoffmann, J.; and Nebel, B. 2005. In defense of pddl axioms. *Artif. Intell.* 168(1):38–69.

To, S. T.; Pontelli, E.; and Son, T. C. 2009. A conformant planner with explicit disjunctive representation of belief states. In *ICAPS*.