# Finding plans subject to stipulations on what information they divulge

Yulin Zhang[1], Dylan A. Shell[1] and Jason M. O'Kane[2]

[1] Texas A&M University, College Station TX, USA
[2] University of South Carolina, Columbia SC, USA

**Abstract.** Motivated by applications where privacy is important, we study planning problems for robots acting in the presence of an observer. We first formulate and then solve planning problems subject to stipulations on the information divulged during plan execution—the appropriate solution concept being both a plan and an information disclosure policy. We pose this class of problem under a worst-case model within the framework of procrustean graphs, formulating the disclosure policy as a particular type of map on edge labels. We devise algorithms that, given a planning problem supplemented with an information stipulation, can find a plan, associated disclosure policy, or both jointly, if and only if some exists. The pair together, comprising the plan and associated disclosure policy, may depend subtly on additional information available to the observer, such as whether the observer knows the robot's plan (e.g., leaked via a side-channel). Our implementation finds a plan and a suitable disclosure policy, jointly, when any such pair exists, albeit for small problem instances.

## 1 Introduction

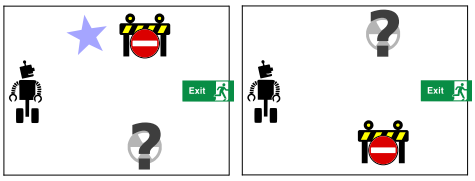
In 2017, iRobot announced that they intended to sell maps of people's homes, as generated by their robot vacuum cleaners. The result was a public outcry [1]. It is increasingly clear that, as robots become part of our everyday lives, the information they could collect (indeed, may *need* to collect to function) can be both sensitive and valuable. Information about a robot's internal state and its estimates of the world's state are leaked by status displays, logged data, actions executed, and information outputted — often what the robot is tasked with doing. The tension between utility and privacy is fundamental.

Typically, robots strive to decrease uncertainty. Some prior work, albeit limited, has illustrated how to cultivate uncertainty, examining how to constrain a robot's beliefs so that it never learns sensitive information (cf. [2,3,4]). In so doing, one precludes sensitive information being disclosed to any adversary. But not disclosing secrets by simply never knowing any, limits the applicability of the approach severely. This paper proposes a more general, wider-reaching model for privacy, beyond mere ingénue robots.

This article posits a potentially adversarial observer and then stipulates properties of what shall be divulged. The stipulation describes information that must be communicated (being required to perform the task) as well as information (confidential information potentially violating the user's privacy) that shouldn't be. Practical scenarios where

This work was supported by NSF awards IIS-1453652, IIS-1527436, and IIS-1526862.

this model applies include: $(i)$ privacy-aware care robots that assist the housebound, providing nursing care; $(ii)$ inspection of sensitive facilities by robots to certify compliance with regulatory agreements, whilst protecting other proprietary or secret details; $(iii)$ sending data remotely to computing services on untrusted cloud infrastructure.




Pebble bed facility Breeder reactor

Fig. 1: **Nuclear Site Inspection** A robot inspects a nuclear facility by taking a measurement at the '**?**' location, which depends on the facility type. But the type of the facility is sensitive information that it must not be divulged to any external observers.

Figure 1 illustrates a scenario which, though simplistic, is rich enough to depict several aspects of the problem. The task requires that a robot determine whether some facility's processing of raw radioactive material meets international treaty requirements or not. The measurement procedure itself depends on the type of facility as the differing physical arrangements of 'pebble bed' and 'breeder' reactors necessitate different actions. First, the robot must actively determine the facility type (checking for the presence of the telltale blue light in the correct spot). Then it can go to a location to make the measurement, the measurement location corresponding with the facility type. But the facility type is sensitive information and the robot must ascertain the radioactivity state while ensuring that the facility type is not disclosed.

What makes this scenario interesting is that the task is rendered infeasible immediately if one prescribes a policy to ensure that the robot never gains sensitive information. Over and above the (classical) question of how to balance information-gathering and progress-making actions, the robot must control what it divulges, strategically increasing uncertainty as needed, precisely limiting and reasoning about the 'knowledge gap' between the external observer and itself. To solve such problems, the robot needs a carefully constructed plan and must establish a policy characterizing what information it divulges, the former achieving the goals set for the robot, the latter respecting all stipulated constraints—and, of course, each depending on the other.

### 1.1 Contributions and itinerary

This paper contributes the first formulation, to our knowledge, of planning where solutions can be constrained so as to require that some information be communicated and other information obscured subject to an adversarial model of an observer. Nor do we know of other work where both a plan and some notion of an interface (the disclosure policy, in our terminology) can both be solved for jointly. The paper is organized as follows: after discussion of related work, Section 3 develops the preliminaries, notation, and formalism, Section 4 addresses an important technical detail regarding an observer's background knowledge, and Section 5 finding plans that satisfy the stipulations. The last section reports experiments conducted with our implementation.

## 2 Related work

An important topic in HRI is expressive action (e.g., see [5]). In recent years there has been a great deal of interest in mathematical models that enable generation of communicative plans. Important formulations include those of [6,7], proposing plausible models for human observers (from the perspectives of presumed cost efficiency, surprisal, or generalizations thereof). In this prior work, conveying information becomes part of an optimization objective, whereas we treat it as a constraint instead. Both [6] and [7] are probabilistic in nature, here we consider a worst-case model that is arguably more suitable for privacy considerations: We ask what an observer can plausibly infer via the history of its received observations. In doing so, we are influenced by the philosophy of LaValle [8], following his use of the term information state (I-state) to refer to a representation of information derived from a history of observations. Finally, since parts of our stipulations may require concealing information, we point out there is also recent work in deception (see [9,10]) and also obfuscation [11].

## 3 The model: worlds, robots and observers

Figure 2 illustrates the three-way relationships underlying the setting we examine. Most fundamentally, a robot executes a *plan* to achieve some goal in the *world*, and the coupling of these two elements generates a stream of observations and actions. Both the plan and the action–observation stream are disclosed, though potentially only partially, to a third party, we term the *observer*. The observer uses the stream, its knowledge of the plan, and also other known structure to infer properties about the interaction. Additionally, a stipulation is provided specifying particular properties that can be learned by the observer. We formalize these elements in terms of p-graphs and label maps (see [12]).

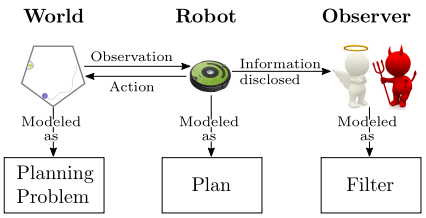



Fig. 2: An overview of the setting: the robot is modeled abstractly as realizing a plan to achieve some goal in the world and a third party observes, modeled as a filter. All three, the world, plan, and filter have concrete representations as p-graphs.

### 3.1 P-graph and its interaction language

We will start with the definition of p-graphs [12] and related properties:

**Definition 1 (p-graph).** *A* p-graph *is an edge-labelled directed bipartite graph with* $G = (V_y \cup V_u, Y, U, V_0)$*, where*

*1) the finite vertex set* $V(G) \coloneqq V_y \cup V_u$*, whose elements are also called* states*, comprises two disjoint subsets: the* observation vertices $V_y$ *and the* action vertices $V_u$*,*

*2) each edge* $e$ *originating at an observation vertex bears a set of observations* $Y(e) \subseteq Y$*, containing* observation labels*, and leads to an action vertex,*

*3) each edge $e$ originating at an action vertex bears a set of actions $U(e) \subseteq U$, containing* action labels*, and leads to an observation vertex, and*

*4) a non-empty set of states $V_0$ are designated as* initial states*, which may be either exclusively action states ($V_0 \subseteq V_u$) or exclusively observation states ($V_0 \subseteq V_y$).*

An *event* is an action or an observation. Respectively, they make up the sets $U$ and $Y$, which are called the p-graph's *action space* and *observation space*. We will also write $Y(G)$ and $U(G)$ for the observation space and action space of $G$. Though that is a slight abuse of notation, the initial states will be written $V_0(G)$, similarly.

Intuitively, a p-graph abstractly represents a (potentially non-deterministic) transition system where transitions are either of type 'action' or 'observation' and these two alternate. The following definitions make this idea precise.

**Definition 2 (transitions to).** *For a given p-graph $G$ and two states $v, w \in V(G)$, a sequence of events $\ell_1, \ldots, \ell_k$* transitions in $G$ from $v$ to $w$ *if there exists a sequence of states $v_1, \ldots, v_{k+1}$, such that $v_1 = v$, $v_{k+1} = w$, and for each $i = 1, \ldots, k$, there exists an edge $v_i \xrightarrow{E_i} v_{i+1}$ for which $\ell_i \in E_i$, and $E_i$ is a subset of $Y(G)$ if $v_i$ is in $V_y$, or a subset of $U(G)$ if $v_i$ is in $V_u$.*

Concisely, we let the predicate $\textsc{TransTo}(v \xrightarrow{s} w)^G$ hold if there is some way of tracing $s$ on $G$ from $v$ to $w$, i.e., it is $\mathrm{True}$ iff $v$ transitions to $w$ under execution $s$. Note, when $G$ has non-deterministic transitions, $v$ may transition to multiple vertices under the same execution. We only require that $w$ be one of them.

**Definition 3 (executions and interaction language).** *An* execution *on a p-graph $G$ is a finite sequence of events $s$, if there exists some $v \in V_0(G)$ and some $w \in V(G)$ for which $\textsc{TransTo}(v \xrightarrow{s} w)^G$. The set of all executions on $G$ is called the* interaction language *(or, briefly, just* language*) of $G$ and is written $\mathcal{L}(G)$.*

Given any edge $e$, if $U(e) = L_e$ or $Y(e) = L_e$, we speak of $e$ *bearing* the set $L_e$.

**Definition 4 (joint-execution).** *A* joint-execution *on two p-graphs $G_1$ and $G_2$ is a sequence of events $s$ that is an execution of both $G_1$ and $G_2$, written as $s \in \mathcal{L}(G_1) \cap \mathcal{L}(G_2)$. The p-graph producing all the joint-executions of $G_1$ and $G_2$ is their tensor product graph with initial states $V_0(G_1) \times V_0(G_2)$, which we denote $G_1 \otimes G_1$.*

A vertex from $G_1 \otimes G_2$ is as a pair $(v_1, v_2)$, where $v_1 \in V(G_1)$ and $v_2 \in V(G_2)$. Next, the relationship between the executions and vertices is established.

**Definition 5.** *The set of vertices reached by execution $s$ in $G$, denoted $\mathcal{V}_s^G$, are the vertices to which the execution $s \in \mathcal{L}(G)$ transitions, starting at an initial state. Symbolically, $\mathcal{V}_s^G \coloneqq \{v \in V(G) \mid \exists v_0 \in V_0(G), \textsc{TransTo}(v_0 \xrightarrow{s} v)^G\}$. Further, the set of executions reaching vertex $v$ in $G$ is written as $\mathcal{S}_v^G \coloneqq \{s \in \mathcal{L}(G) \mid v \in \mathcal{V}_s^G\}$.*

The naming here serving to remind that $\mathcal{V}$ describes sets of vertices, $\mathcal{S}$ describes sets of strings/executions. The collection of sets $\{\mathcal{S}_{v_0}^G, \mathcal{S}_{v_1}^G, \ldots, \mathcal{S}_{v_i}^G \ldots\}$ can be used to form an equivalence relation $\underset{G}{\sim}$ over executions, under which $s_1 \underset{G}{\sim} s_2$ if and only if $\mathcal{V}_{s_1}^G = \mathcal{V}_{s_2}^G$. This equivalence relation partitions the executions in $\mathcal{L}(G)$ into a set of non-empty equivalence classes: $\mathcal{L}(G)/\underset{G}{\sim} = \{[r_0]_G, [r_1]_G, [r_2]_G, \ldots\}$, where each

equivalence class is $[r_i]_G = \{s \in \mathcal{L}(G) \,|\, r_i \underset{G}{\sim} s\}$ and $r_i$ is a representative execution in $[r_i]_G$. The intuition is that any two executions that transition to identical sets of vertices are, in an important sense, indistinguishable.

We shall consider systems where the vertices of a p-graph constitute the state that is stored, acted upon, and/or represented—they are, thus, akin to a 'sufficient statistic'.

**Definition 6 (state-determined).** *A p-graph $G$ is in a* state-determined *presentation, or is in* state-determined *form, if* $\forall s \in \mathcal{L}(G), |\mathcal{V}_s^G| = 1$.

The procedure to expand any p-graph $G$ into a state-determined presentation $\textsc{Sde}(G)$ can be found in Algorithm 2 of [12]. The language of p-graphs is not affected by state-determined expansion, i.e., $\mathcal{L}(G) = \mathcal{L}(\textsc{Sde}(G))$.

Next, one may start with vertices and ask about the executions reaching those vertices. (Later, this will be part of how an observer makes inferences about the world.)

**Definition 7.** *Given any set of vertices $B \subseteq V(G)$ in p-graph $G$, the set of executions that reach exactly (i.e. reach and reach only) $B$ is* $\mathbb{S}_B^G := (\cap_{v\in B}\mathcal{S}_v^G) \setminus \cup_{v\in(V(G)\setminus B)}\mathcal{S}_v^G$.

Above, the $\cap_{v\in B}\mathcal{S}_v^G$ represents the set of executions that reach every vertex in $B$. By subtracting the ones that also reach the vertices outside $B$, $\mathbb{S}_B^G$ describes the set of executions that reach exactly $B$. In Figure 3, the executions reaching $w_3$ are represented as $\mathcal{S}_{w_3}^G = \{a_1o_1, a_2o_1\}$. But the executions reaching and reaching only $\{w_3\}$ are $\mathbb{S}_{\{w_3\}}^G = \{a_1o_1\}$ since $a_2o_1$ also reaches $w_4$. Specifically, the equivalence class $[r_i]_G$ contains the executions that reach exactly $\mathcal{V}_{r_i}^G$, so we have $[r_i]_G = \mathbb{S}_{\mathcal{V}_{r_i}^G}^G$.

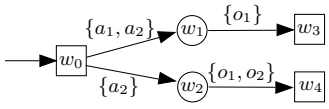



Fig. 3: An example showing the difference between 'reaches' and 'reaches exactly' as distinguished in notation as $\mathcal{S}_w^G$ and $\mathbb{S}_{\{w\}}^G$.

### 3.2 Planning problems and plans

In the p-graph formalism, planning problems and plans are defined as follows [12].

**Definition 8 (planning problems and plans).** *A* planning problem *is a p-graph $W$ along with a goal region $V_{\text{goal}} \subseteq V(W)$; a* plan *is a p-graph $P$ equipped with a termination region $V_{\text{term}} \subseteq V(P)$.*

Planning problem $(W, V_{\text{goal}})$ is solved by some plan $(P, V_{\text{term}})$ if the plan always terminates (i.e., reaches $V_{\text{term}}$) and only terminates at a goal. Said with more precision:

**Definition 9 (solves).** *A plan $(P, V_{\text{term}})$* solves *a planning problem $(W, V_{\text{goal}})$ if there is some integer which bounds length of all joint-executions, and for each joint-execution and any pair of nodes $(v \in V(P), w \in V(W))$ reached by that execution simultaneously, the following conditions hold:*

*1) if $v$ and $w$ are both action nodes and, for every label borne by each edge originating at $v$, there exist edges originating at $w$ bearing the same action label;*

2) *if $v$ and $w$ are both observation nodes and, for every label borne by each edge originating at $w$, there exist edges originating at $v$ bearing the same observation label;*
3) *if $v \in V_{\text{term}}$ and then $w \in V_{\text{goal}}$;*
4) *if $v \notin V_{\text{term}}$ then some extended joint-execution exists, continuing from $v$ and $w$, that does reach the termination region.*

In the above, properties 1) and 2) describe a notion of safety; property 3) of correctness; and 4) of liveness. In the previous definition, there is an upper bound on joint-execution length. We say that plan $(P, V_{\text{term}})$ is *c-bounded* if, $\forall s \in \mathcal{L}(P)$, $|s| \leq c$.

### 3.3 Information disclosure policy, divulged plan, and observer

The agent who is the observer sees a stream of the robot's actions and observations, and uses them to build estimates (or to compute general properties) of the robot's interaction with the world. But the observer's access to this information will usually be imperfect—either by design, as a consequence of real-world imperfections, or some combination of both. Conceptually, this is a form of partial observability in which the stream of symbols emitted as part of the robot's execution is distorted into to the symbols seen by the observer (see Figure 4). For example, if some pairs of actions are indistinguishable from the perspective of the observer, this may be expressed with a function that maps those pairs of actions to the same value. In this paper, this barrier is what we have been referring to (informally, thus far) with the phrase *information disclosure policy*. It is formalized as a mapping from the events in the robot's true execution in the world p-graph to the events received by the observer.

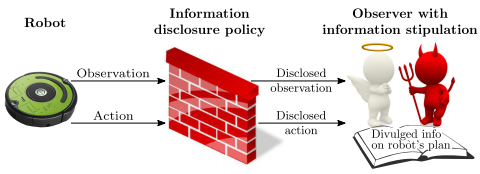



Fig. 4: The information disclosure policy, divulged plan and information stipulation. Even when the observer is a strong adversary, the disclosure policy and divulged plan can limit the observer's capabilities effectively.

**Definition 10 (Information disclosure policy).** *An* information disclosure policy *is a label map $h$ on p-graph $G$, mapping from elements in the combined observation and action space $Y(G) \cup U(G)$ to some set of events $X$.*

The word 'policy' hints at two interpretations: first, as something given as a predetermined arrangement (that is, as a rule); secondly, as something to be sought (together with a plan). Both senses apply in the present work; the exact transformation describing the disclosure of information will be used first (in Section 5.1) as a specification and then, later (in Section 5.2) as something which planning algorithms can produce. How the information disclosure policy is realized in some setting depends on which sense is apt: it can be interpreted as describing observers (showing that for those observers unable to tell $y_i$ from $y_j$, the stipulations can be met), or it can inform robot operation (the stipulations require that the robot obfuscate $u_\ell$ and $u_m$ via means such as explicit concealment, sleight-of-hand, misdirection, etc.)

The observer, in addition, may also have imperfect knowledge of robot's plan, which is leaked or communicated from the side-channel. The *disclosed plan* is also modeled as a p-graph, which may be weaker than knowing the actual plan. A variety of different types of divulged plan are introduced later (in Section 4) to model different prior knowledge available to an observer; as we will show, despite their differences, they can be treated in a single unified way.

The next step is to provide formal definitions for the ideas just described. In the following, we refer to $h$ as the map from the set $Y \cup U$ to some set $X$, and refer to its preimage $h^{-1}$ as the map from $X$ to subsets of $Y \cup U$. The notation for a label map $h$ and its preimage $h^{-1}$ is extended in the usual way to sequences and sets: we consider sets of events, executions (being sequences), and sets of executions. They are also extended to p-graphs in the obvious way, by applying the function to all edges.

For brevity's sake, the outputs of $h$ will be referred to simply as 'the image space.' The function $h$ may either preserve information (when a bijection) or lose information (with multiple inputs mapped to one output). The loss of information is felt in $Y \cup U$ by the extent to which some element of $Y \cup U$ grows under $h^{-1} \circ h$, and for all $\ell \in Y \cup U$, $h^{-1} \circ h(\ell) \supseteq \{\ell\}$. In contrast, starting from $x \in X$, the uncertainty, apparent via set cardinality under $h^{-1}$, is washed out again when pushed forward to the image space $X$ via $h \circ h^{-1}$, i.e., $\forall x \in X$, $h \circ h^{-1}(x) = \{x\}$.

**Definition 11 (I-state graph).** *For planning problem $(W, V_{\mathrm{goal}})$, plan $(P, V_{\mathrm{term}})$ and information disclosure policy $h : Y(W) \cup U(W) \to X$, an observer's* I-state graph *$I$ is a p-graph, whose inputs are from the image space of $h$ (i.e., $Y(I) \cup U(I) = X$), with $\mathcal{L}(I) \supseteq h[\mathcal{L}(W)]$. The action space and observation space of $I$ are also written as $X_u = U(I)$ and $X_y = Y(I)$.*

Inherited from the property of $h \circ h^{-1}$, for any I-state graph $I$, we have $I = h \circ h^{-1}\langle I \rangle$, and $\forall B \subseteq V(I), h^{-1}[\mathbb{S}^I_B] = \mathbb{S}^{h^{-1}\langle I \rangle}_B$.

The observer's I-state graph is a p-graph with events in the image space $X$. By having $\mathcal{L}(I) \supseteq h[\mathcal{L}(W)]$, we are requiring that strings generated in the world can be safely traced on $I$.

Next, we formalize the crucial connection from the interaction of the robot and world, via the stream of symbols generated, to the state tracked by the observer. Inference proceeds from the observer back to the world, though causality runs the other way (glance again at Figure 2). We begin, accordingly, with that latter direction.

**Definition 12 (compatible world states).** *Given observer I-state graph $I$, robot's plan $(P, V_{\mathrm{term}})$, world graph $(W, V_{\mathrm{goal}})$, and label map $h$, the world state $w$ is* compatible *with the set of I-states $B \subseteq V(I)$ if $\exists s \in \mathcal{L}(W)$ such that $s \in \underbrace{h^{-1}[\mathbb{S}^I_B]}_{(1)} \cap \underbrace{\mathcal{L}(P)}_{(2)} \cap \underbrace{\mathcal{S}^W_w}_{(3)}$.*

Informally, each of the three terms can be interpreted as:

(1) An observer with I-state graph $I$ may ask which sequences are responsible for having arrived at states $B$. The answer is the set $\mathbb{S}^I_B$, being the executions contained in equivalence classes that are indistinguishable up to states in $I$. Those strings are in the image space $X$, so, to obtain an answer in $Y \cup U$, we take their preimages. Every execution in $h^{-1}[\mathbb{S}^I_B]$ leads the observer to $B$. Note that information may be degraded by either $h$, $I$, or both.

(2) The set of executions that may be executed by the robot is represented by $\mathcal{L}(P)$. If the observer knows that the robot's plan takes, say, the high road, this information allows the observer to remove executions involving the robot along the low road.
(3) The set of executions reaching world state $w$ is represented by $\mathcal{S}_w^W$. Two world states $w, w' \in V(W)$ are essentially indiscernible or indistinguishable if $\mathcal{S}_w^W = \mathcal{S}_{w'}^W$, as the sets capture the intrinsic uncertainty of world $W$.

When an observer is in $B$, and $w$ is compatible with $B$, there exists some execution, a certificate, that the world could plausibly be in $w$ subject to (1) the current information summarized in $I$; (2) the robot's plan; (3) the structure of the world. The set of *all* world states that are compatible with $B$ is denoted $\mathcal{W}_B^{I,P}$, which is the observer's estimate of the world states when known information about $W$, $P$ and $I$ have all been incorporated.

A typical observer may know less about the robot's future behavior than the robot's full plan. Weaker knowledge of how the robot will behave can be expressed in terms of some p-graph $D$, such that $\mathcal{L}(D) \supseteq \mathcal{L}(P)$. (Here the mnemonic is that it is the divulged information about the robot's plan, which one might imagine as leaked or communicated via a side-channel.) Notice that the information divulged to the observer about the robot's execution is in the preimage space. The key reason for this modeling decision is that information may be lost under label map $h$; an observer gains the greatest information when the plan is disclosed in the preimage space and, as we consider worst-case conditions, we are interested in what the strongest (even adversarial) observers might infer. Thus, we study divulgence where the observer obtains as much as possible.

Definition 12 requires the substitution of the second term in the intersection with $\mathcal{L}(D)$. When only $D$ is given, the most precise inference replaces $\mathcal{W}_B^{I,P}$ with $\mathcal{W}_B^{I,D}$:

**Definition 13 (estimated world states).** *Given an I-state graph $I$, divulged plan p-graph $D$, world p-graph $W$, and label map $h$, the set of estimated world states for I-states $B \subseteq V(I)$ is* $\mathcal{W}_B^{I,D} \coloneqq \left\{ w \in V(W) \;\middle|\; (\mathbb{S}_B^{h^{-1}\langle I \rangle} \cap \mathcal{L}(D) \cap \mathcal{S}_w^W) \neq \varnothing \right\}$.

Observe that $h^{-1}[\mathbb{S}_B^I]$ has been replaced with $\mathbb{S}_B^{h^{-1}\langle I \rangle}$, since $h^{-1}[\mathbb{S}_B^I] = \mathbb{S}_B^{h^{-1}\langle I \rangle}$.

The last remaining element in Figure 4 that needs to be addressed is the stipulation of information. We do that next.

### 3.4 Information stipulations

We prescribe properties of the information that an observer may extract from its input by imposing constraints on the sets of estimated world states. The observer, filtering a stream of inputs sequentially, forms a correspondence between its I-states and world states. We write propositional formulas with semantics defined in terms of this correspondence—in this model the stipulations are written to hold over *every* reachable set of associated states.[3]

First, however, we must delineate the scope of the estimated world states to be constrained. Some states, in inherently non-deterministic worlds, may be inseparable because they are reached by the same execution. In Figure 3, both $w_3$ and $w_4$ will be

[3] We foresee other variants which are straightforward to modifications to consider; but we report only on our current implementation.

$$\begin{aligned}
&\text{Formula} \to \text{Clause}_1 \wedge \ldots \wedge \text{Clause}_n \\
&\text{Clause} \to \text{Literal}_1 \vee \ldots \vee \text{Literal}_m \\
&\text{Literal} \to \text{Symbol} \mid \neg\text{Symbol} \\
&\text{Symbol} \to v_0, v_1, v_2, \ldots
\end{aligned}$$

$$\frac{\text{[VALUE]}}{\langle v_i \rangle \Downarrow \text{eval}(v_i \overset{?}{\in} \mathcal{W}_B^{I,D})} \qquad \frac{\text{[NOT]} \quad \langle v_i \rangle \Downarrow w}{\langle \neg v_i \rangle \Downarrow \text{the negation of } w}$$

$$\frac{\text{[OR]} \quad \langle \ell_1 \rangle \Downarrow w_1 \quad \langle \ell_2 \rangle \Downarrow w_2}{\langle \ell_1 \vee \ell_2 \rangle \Downarrow \text{the logical or of } w_1 \text{ and } w_2} \qquad \frac{\text{[AND]} \quad \langle c_1 \rangle \Downarrow w_1 \quad \langle c_2 \rangle \Downarrow w_2}{\langle c_1 \wedge c_2 \rangle \Downarrow \text{the logical and of } w_1 \text{ and } w_2}$$

Fig. 5: The syntax and natural semantics of the information stipulations, where $c_i$, $\ell_i$, $v_i$, represent a clause, literal, and symbol, respectively, and $w_i$ is the result of the evaluation. The transition $\langle e \rangle \Downarrow w$ denotes a transition, where $e$ is any expression defined by the grammar and $w$ is the value yielded by the expression.

reached (non-deterministically) by execution $a_2 o_1$. Since this is intrinsic to the world, even when the observer has perfect observations, they remain indistinguishable. In the remainder of this paper, we will assume that the world graph $W$ is in state-determined form, and we may affix stipulations to the world states knowing that no two vertices will be non-deterministically reached by the same execution.

Second, we write propositional formulae to constrain the observer's estimate. Formula $\boldsymbol{\Phi}$ is written in conjunctive normal form, consisting of symbols, literals and clauses as shown in Fig. 5. Firstly, an atomic symbol $v_i$ is associated with each world state $v_i \in V(W)$. If $v_i$ is contained in the observer's estimates $\mathcal{W}_B^{I,D}$, we will evaluate the corresponding symbol $v_i$ as $\mathrm{True}$. It evaluates as $\mathrm{False}$ otherwise. With each symbol grounded in this way, we evaluate literals and clauses compositionally, using logic operators NOT, AND, OR. These are defined in the standard way, eventually enabling evaluation of $\boldsymbol{\Phi}$ on the observer's estimate $\mathcal{W}_B^{I,D}$.

Let the predicate $\mathrm{satfd}(B, \boldsymbol{\Phi})$ denote whether the stipulation $\boldsymbol{\Phi}$ holds for I-states $B$. Then a plan $P$ satisfies the stipulations, if and only if

$$\forall s \in \mathcal{L}(P) \cap \mathcal{L}(W)\ B = \mathcal{V}_{h(s)}^{I}\ \ \mathrm{satfd}(B, \boldsymbol{\Phi}).$$

## 4 The observer's knowledge of the robot's plan

Above, we hinted that observers may differ depending on the prior knowledge that has been revealed to them; next we bring this idea into sharper focus. The information associated with an observer is contained in a pair $(I, D)$: the I-state graph $I$ that acts as a filter, succinctly tracking state from a stream of inputs, and knowledge of robot's plan in the form of a p-graph $D$. These two elements, through Definition 13, allow the observer to form a correspondence with the external world $W$. The I-state graph $I$ induces $\underset{I}{\sim}$ over its set of executions and hence over the joint-executions with the world, or, more precisely, the image of those through $h$. By comparing the fineness of the relations induced by two I-state graphs, one obtains a sense of the relative coarseness of the two I-state graphs. As the present paper describes methods motivated by applications to robotic privacy, we model the most capable adversary, taking the *finest observer*, that is, one whose equivalence classes are as small as possible.

**Definition 14 (finest observer).** *Given world graph $W$ and the divulged plan $D$, an I-state graph $\widetilde{I}$ is a* finest observer *if for any I-state graph $I$, we have $\forall s \in \mathcal{L}(W)$, $\mathcal{W}^{\widetilde{I},D}_{h(s)} \subseteq \mathcal{W}^{I,D}_{h(s)}$.*

**Lemma 1.** *$h\langle W\rangle$ is a finest observer.*

By way of a proof sketch, note that the observer only ever sees the image of the world under the label map $h$, i.e. $h\langle W\rangle$. The p-graph $h\langle W\rangle$ serves as a natural I-state graph for a finest observer as it allows the observer to have sufficient internal structure to keep track of every world state.

The second element in the observer pair is $D$, information disclosed about the plan, and presumed to be known *a priori*, to the observer. Depending on how much the observer knows, there are multiple possibilities here, from most- to least-informed:

I. The observer knows the exact plan $P$ to be executed.
II. The plan to be executed is among a finite collection of plans $\{P_1, P_2, \ldots, P_n\}$.
III. The observer may only know that the robot is executing *some* plan, that is, the robot is goal directed and aims to achieve some state in $V_{\text{goal}}$.
IV. The observer knows nothing about the robot's execution other than that it is on $W$.

It turns out that a p-graph exists whose language expresses knowledge for each of those cases (we omit the details here). Furthermore, Section 3.3 details how the observer's knowledge of the world state ($\mathcal{W}^{I,D}_B$) from I-states $B$ depends on $\mathbb{S}_B^{h^{-1}\langle I\rangle} \cap \mathcal{L}(D) \cap \mathcal{L}(W)$, a set of executions that arrive at $B$ in the I-state graph $I$. Because the observer uses $D$ to refine $\mathbb{S}_B^{h^{-1}\langle I\rangle}$, when $\mathcal{L}(P) \subsetneq \mathcal{L}(D)$ the gap between the two sets of executions represents a form of uncertainty. The ordering of the four cases, thus, can be stated precisely in terms of language inclusion.

Now using the $D$ as appropriate for each case, one may examine whether a given plan and disclosure policy solves the planning problem (i.e., achieves desired goals in the world) while meeting the stipulations on information communicated. Hence, we see that describing disclosed information via a p-graph $D$ is in fact rather expressive. This section has also illustrated the benefits of being able to use both interaction language and graph presentation views of the same structure.

## 5 Searching for plans and disclosure policy: the SEEK problems

In this section, we will show how to search for a plan (together with the label map).

**Problem:** $\textsc{Seek}_{\boldsymbol{x}}\big((W, V_{\text{goal}}), \boldsymbol{x}, (\widetilde{I}, D), h, \boldsymbol{\Phi}\big)$
$\textsc{Seek}_{\boldsymbol{x},\boldsymbol{\lambda}}\big((W, V_{\text{goal}}), \boldsymbol{x}, (\widetilde{I}, \boldsymbol{x}), \boldsymbol{\lambda}, \boldsymbol{\Phi}\big)$

Vars. to solve for:
$\boldsymbol{x}$ is a plan
$\boldsymbol{\lambda}$ is a label map

*Input:* A planning problem $(W, V_{\text{goal}})$, a finest observer $\widetilde{I}$, a divulged plan p-graph $D$, information disclosure policy $h$ and information stipulation $\boldsymbol{\Phi}$.

*Output:* A plan $\boldsymbol{x} = (P, V_{\text{term}})$ and/or label map $\boldsymbol{\lambda} = h$ such that plan $(P, V_{\text{term}})$ solves the problem $(W, V_{\text{goal}})$, and $\forall s \in \mathcal{L}(W^\dagger) \cap \mathcal{L}(P), B = \mathcal{V}^I_{h(s)}$, the information stipulation $\boldsymbol{\Phi}$ is always evaluated as True on $\mathcal{W}^{I,D}_B$ (i.e. $\text{satfd}(B, \boldsymbol{\Phi}) = \text{True}$), else False.

Of the two versions of SEEK, the first searches for a plan, the second for a plan and a label map, jointly. We consider each in turn.

### 5.1 Finding a plan given some predetermined $D$

For $\text{SEEK}_{\boldsymbol{x}}$, first we must consider the search space of plans. Prior work [12] showed that, although planning problems can have stranger solutions than people usually contemplate, there is a core of well-structured plans (called homomorphic solutions) that suffice to determine solvability. As an example, there may exist plans which loop around the environment before achieving the goal, but, they showed that in seeking plans, one need only consider plans that short-circuit the loops.

The situation is rather different when a plan must satisfy more than mere goal achievement: information stipulations may actually *require* a plan to loop in order to ensure that the disclosed stream of events is appropriate for the observer's eyes. (A concrete example appears in Fig. 7(c).) The argument in [12] needs modification for our problem—a different construction can save the result even under disclosure constraints. This fact is key to be able to implement a solution.

In this paper, without loss of generality, we focus on finding plans in state-determined form. Next, we will examine the solution space closely.

**Definition 15.** *A plan $P$ is* congruent *on the world graph $W$, if and only if for every pair of executions $s_1, s_2 \in \mathcal{L}(P)$ we have $s_1 \underset{P}{\sim} s_2 \implies s_1 \underset{W}{\sim} s_2$.*

In other words, a plan that respects the equivalence classes of the world graph is defined as a congruent plan. Next, our search space is narrowed further still.

**Lemma 2.** *Given any plan $(P, V_{\text{term}})$, there exists a plan $(P', V'_{\text{term}})$ that is congruent on the world graph $W$ and $\mathcal{L}(P') = \mathcal{L}(P)$.*

*Proof.* We give a construction from $P$ of $P'$ as a tree, and show that it meets the conditions. To construct $P'$, perform a BFS on $P$. Starting from $V_0(P)$, build a starting vertex $v_0$ in $P'$, keep a correspondence between it and $V_0(P)$. Mark $v_0$ as unexpanded. Now, for every unexpanded vertex $v$ in $P'$, mark the set of all outgoing labels for its corresponding vertices in $P$ as $L_v$, create a new vertex $v'$ in $P'$ for each label $l \in L_v$, build an edge from $v$ to $v'$ with label $l$ in $P'$, and mark it as expanded. Repeat this process until all vertices in $P'$ have been expanded. Mark the vertices corresponding to vertices in $V_{\text{term}}$ as $V'_{\text{term}}$. In the new plan $(P', V'_{\text{term}})$, no two executions reach the same vertex. That is, $\forall s_1, s_2 \in \mathcal{L}(P'), s_1 \not\underset{P'}{\sim} s_2$. Hence, $P'$ is congruent on $W$. In addition, since no new executions are introduced and no executions in $P$ are eliminated during the construction of $P'$, we have $\mathcal{L}(P') = \mathcal{L}(P)$. □

**Theorem 1.** *For problem $\text{SEEK}_{\boldsymbol{x}}\big((W, V_{\text{goal}}), \boldsymbol{x}, (I, D), h, \boldsymbol{\Phi}\big)$, if there exists a solution $(P, V_{\text{term}})$, then there exists a solution $(P', V'_{\text{term}})$ that is both $c$-bounded and congruent on $W$, where $c = |V(W)| \cdot |V(D)| \cdot |V(I)|$.*

*Proof.* Suppose $\text{SEEK}_{\boldsymbol{x}}$ has a solution $(P, V_{\text{term}})$. Then the existence of a solution $(P', V'_{\text{term}})$ which is congruent on $W$ is implied by Lemma 2. Moreover, we have $\text{CHECK}\big((W, V_{\text{goal}}), (P, V_{\text{term}}), D, I, h, \boldsymbol{\Phi}\big) \implies \text{CHECK}\big((W, V_{\text{goal}}), (P', V'_{\text{term}}), D, I, h, \boldsymbol{\Phi}\big)$, following from two observations:

(*i.*) if $(P, V_{\text{term}})$ solves $(W, V_{\text{goal}})$ then the means of construction ensures $(P', V'_{\text{term}})$ does as well, and

(*ii.*) in checking $\boldsymbol{\Phi}$, the set of estimated world states $\mathcal{W}^{I,D}_{\{v\}}$ does not change for each vertex $v \in V(\text{SDE}(I))$, since the triple graph is independent of the plan to be searched. The set of I-states to be evaluated by $\boldsymbol{\Phi}$ in $\text{SDE}(I)$ is $\cup_{s' \in h[\mathcal{L}(P) \cap \mathcal{L}(W)]} \mathcal{V}^{\text{SDE}(I)}_{s'}$. Since $\mathcal{L}(P) = \mathcal{L}(P')$, the set of I-states to be evaluated is no altered and the truth of $\boldsymbol{\Phi}$ along the plan is preserved.

The final step is to prove that if there exists a congruent solution $(P', V'_{\text{term}})$, then there exits a solution $(P'', V''_{term})$ that is $c$-bounded. First, build a product graph $T$ of $W$, $D$, and $h^{-1}\langle SED(I)\rangle$, with vertex set $V(W) \times V(D) \times V(h^{-1}\langle \text{SDE}(I)\rangle)$. Then trace every execution $s$ in $P'$ on $T$. If $s$ visits the same vertex $(v^W, v^D, v^{h^{-1}\langle \text{SDE}(I)\rangle})$ multiple times, then $v^W$, $v^D$, and $v^{h^{-1}\langle \text{SDE}(I)\rangle}$ have to be action vertices, for otherwise $P'$ can loop forever and is not a solution (since $P'$ is finite on $W$). Next, record the action taken at the last visit of $(v^W, v^P, v^{h^{-1}\langle \text{SDE}(I)\rangle})$ as $a_{\text{last}}$. Finally, build a new plan $(P'', V'_{\text{term}})$ by bypassing unnecessary transitions on $P'$ as follows. For each vertex $(v^W, v^P, v^{h^{-1}\langle \text{SDE}(I)\rangle})$ that is visited multiple times, $P''$ takes action $a_{\text{last}}$ when $(v^W, v^P, v^{h^{-1}\langle \text{SDE}(I)\rangle})$ is first visited. $P''$ terminates at the goal states without violating any stipulations, since it takes a shortcut in the executions of $P'$ but—crucially—without visiting any new observer I-states. In addition, $P''$ will visit each vertex in $T$ at most once, and the maximum length of its executions is $|V(W)| \times |V(D)| \times |V(h^{-1}\langle \text{SDE}(I)\rangle)|$. Since $P''$ preserves the structure of $P'$ during this construction, $P''$ is also congruent. □

The intuition, and the underlying reason for considering congruent plans, is that modifying the plan will not affect the stipulations if the underlying languages are preserved. The bound on the length then takes this further, modifying the language by truncating long executions in the triple graph, thereby shortcutting visits to I-states that do not affect goal achievement.

Accordingly, it suffices to look for congruent plans in the (very specific) form of trees, since any plan has a counterpart that is congruent and in the form of a tree (see Lemma 2 for detail). Theorem 1 states that the depth of the tree is at most $c = |V(W)| \cdot |V(D)| \cdot |V(h^{-1}\langle \text{SDE}(I)\rangle)|$. Therefore, we can limit the search space to trees of a specific bounded depth. To search for a $c$-bounded solution, first we mark the vertex $(v^W, v^D, v^{h^{-1}\langle \text{SDE}(I)\rangle})$ as: (i) a goal state if $v^W$ is a goal state in the world graph; (ii) as satisfying $\boldsymbol{\Phi}$ when all the world states appearing together with $v^{h^{-1}\langle \text{SDE}(I)\rangle}$ together satisfy $\boldsymbol{\Phi}$. Then we will conduct an AND–OR search [13] on the triple graph:

- Each action vertex serves as an OR node, and an action should be chosen for the action vertex such that it will eventually terminate at the goal states and all the vertices satisfy $\boldsymbol{\Phi}$ along the way.
- Each observation vertex is treated as an AND node, and there exists a plan that satisfies $\boldsymbol{\Phi}$ for all its outgoing observation vertices.

### 5.2 Search for plan and label map for the finest observer, disclosing the same

It is not merely the joint search that makes this, the second problem more interesting. Whereas the first has a divulged plan $D$ that is *a priori* fixed, the second uses $\boldsymbol{x}$, the plan that was found, as $D$. This latter fact makes the third substantially more difficult.

At a high level, it is not hard to see why: the definitions in the previous section show that both $P$ and $D$ play a role in determining whether a plan satisfies a stipulation. Where $D$ is known and fixed beforehand (for example, in Case IV, $D = W$, or Case III, $D = P^*$), a solution can proceed by building a correspondence in the triple graph $W \otimes D \otimes h^{-1}\langle \textsc{Sde}(I) \rangle$ and searching in this graph for a plan. In $\textsc{Seek}_{\boldsymbol{x},\boldsymbol{\lambda}}$, however, one is interested in the case where $D = P$, where the divulged plan is tight, being the robot's plan exactly. We cannot search in the same product graph, because we can't make the correspondence since $D$ has yet to be discovered, being determined only after $P$ has been found. Crucially, the feasibility of $P$ depends on $D$, that is, on itself! Finding such a solution requires an approach capable of building incremental correspondences from partial plans. A key result of this paper is that $\textsc{Seek}_{\boldsymbol{x},\boldsymbol{\lambda}}$ is actually solvable without resorting to mere generate-and-check.

**Lemma 3.** *Let $\mathcal{W}$ be estimated world states for the finest observer $h\langle W \rangle$, and let $w$ be the world state which is observable to the robot. If there exists a solution for $\textsc{Seek}_{\boldsymbol{x},\boldsymbol{\lambda}}$, then there exists a solution that only visits each pair $(w, \mathcal{W})$ at most once.*

*Proof.* Let $(P, V_{\text{term}})$ and $h$ be a solution for $\textsc{Seek}_{\boldsymbol{x},\boldsymbol{\lambda}}$. Suppose $P$ visited $(w, \mathcal{W})$ n times. Let the set of actions taken at $i$-th visit be $A_i$. Then we can construct a new plan $(P', V_{\text{term}})$ which always takes $A_n$ at $(w, \mathcal{W})$. If $P$ does not violate the stipulations, then $P'$ will never do since $P'$ is a shortcut of $P$ and never visits more I-states than $P$ does. In addition, $P'$ will also terminate at the goal region if $P$ does. □

**Theorem 2.** *If there exists a solution for $\textsc{Seek}_{\boldsymbol{x},\boldsymbol{\lambda}}\big((W, V_{\text{goal}}), \boldsymbol{x}, (I_f, \boldsymbol{x}), \boldsymbol{\lambda}, \boldsymbol{\Phi}\big)$. then there exists a plan $P$ that takes $(w, \mathcal{W}_B^{h\langle W \rangle, P})$ as its plan state, where $w$ is the world state and the set $\mathcal{W}_B^{h\langle W \rangle, P}$ consists of the estimated world states for I-states $B$. Furthermore, if $(w, \mathcal{W}_B^{h\langle W \rangle, P}) \in V(P)$, then $\forall w' \in \mathcal{W}_B^{h\langle W \rangle, P}$, $(w', \mathcal{W}_B^{h\langle W \rangle, P}) \in V(P)$.*

*Proof.* Lemma 3 shows that we can treat $(w, \mathcal{W}_B^{h\langle W \rangle, P})$ as the plan state for the plan to be searched for.

Since $w' \in \mathcal{W}_B^{h\langle W \rangle, P}$, we have $\exists s \in \mathcal{S}_{w'}^W \cap \mathcal{L}(P) \cap h^{-1}[\mathbb{S}_B^{h\langle W \rangle}]$. Since $s \in \mathcal{L}(P)$, $s$ reaches $w$ and $h(s)$ reaches $B$, we have $s$ reaches the tuple $(w', \mathcal{W}_B^{h\langle W \rangle, P})$. Hence, $(w', \mathcal{W}_B^{h\langle W \rangle, P}) \in V(P)$. □

In searching for $(P, V_{\text{term}})$, for any action state $v_p = (w, \mathcal{W}_B^{h\langle W \rangle, P})$, we determine:

- $w \in V_{\text{goal}}$ : We must decide whether $v_p \in V_{\text{term}}$ holds or not;
- $w \notin V_{\text{goal}}$ : We must choose the set of nonempty actions to be taken at $v_p$. It has to be a set of actions, since these chosen actions are not only aiming for the goal but also obfuscating each other under the label map.

A state $v_p = (w, \mathcal{W}_B^{h\langle W\rangle,P})$ is a terminating state in the plan when $\mathcal{W}_B^{h\langle W\rangle,P} \subseteq V_{\text{goal}}$.

With action choices for each plan state $(w, \mathcal{W}_B^{h\langle W\rangle,P})$ and label map $h$, we are able to maintain transitions of the estimated world states for $B'$ after observing the image $x$. Now, if $(w, \mathcal{W}_B^{h\langle W\rangle,P})$ is an action state, let the set of actions taken at $w$ be $A_w$. Then the label map $h$ partitions the actions in $\cup_{w\in\mathcal{W}_B^{h\langle W\rangle,P}} A_w$ into groups, each of which shares the same image. The estimated worlds states for $B'$ transition in terms of groups

$$\mathcal{W}_{B'}^{h\langle W\rangle,P} = \Big\{ w' \in V(W) \Big| (w, \mathcal{W}_B^{h\langle W\rangle,P}) \notin V_{\text{term}}, w \in \mathcal{W}_B^{h\langle W\rangle,P}, \\ \exists a \in A_w, h(a) = x, \textsc{TransTo}(w \xrightarrow{a} w')^W \Big\}.$$

Conversely, if $(w, \mathcal{W}_B^{h\langle W\rangle,P})$ is an observation state, let the observations available at $w$ be $O_w$. Then $h$ also partitions the observations in $\cup_{(w,\mathcal{W}_B^{h\langle W\rangle,P})\notin V_{\text{term}}} O_w$ and estimated world states for $B'$ transition as

$$\mathcal{W}_{B'}^{h\langle W\rangle,P} = \Big\{ w' \in V(W) \Big| (w, \mathcal{W}_B^{h\langle W\rangle,P}) \notin V_{\text{term}}, w \in \mathcal{W}_B^{h\langle W\rangle,P}, \\ \exists o' \in O_w, h(o') = x, \textsc{TransTo}(w \xrightarrow{o} w')^W \Big\}.$$

Instead of searching for the label map over the set of all actions and observations in $W$, we will first seek a partial label map for all observations or chosen actions for world states in $\mathcal{W}_B^{h\langle W\rangle,P}$, and then incrementally consolidate them. Each partial label map is a partition of the events, making it easy to check whether two partial maps conflict when they are consolidated. If two partial partitions disagree on a value, we backtrack in the search to try another partition label map. Putting it all together as detailed in Fig. 6, we can build a type of AND–OR search tree to incorporate these choices.

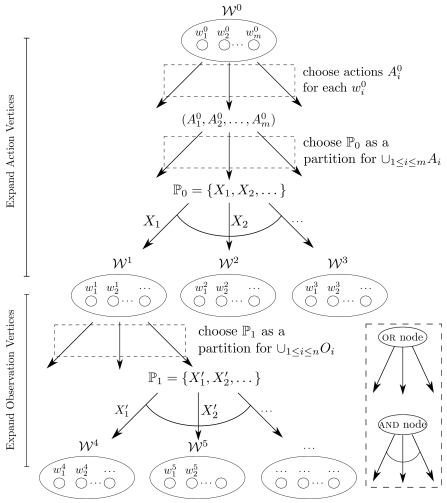



Fig. 6: Solving the $\textsc{Seek}_{\boldsymbol{x},\boldsymbol{\lambda}}$ problem via generalized AND–OR search.

For a set of actions comprising a vertex $\mathcal{W}^0$ two tiers of OR nodes are generated. The first is over subsets $(A_1^0, A_2^0, \dots, A_m^0)$, being possible actions to the take; the second chooses specific partitions of values $\mathbb{P}_i = \{X_1, X_2, \dots\}$, (i.e., partial label maps). A given partition is expanded as an AND node with each outgoing edge bearing a group of events sharing the same image under the partial label map.

Observation vertices $\mathcal{W}^1$ are expanded in a similar way, but are simpler since we forgo the step involving choosing actions.

If there exists a plan and label map then for each $\mathcal{W}$ in the tree, there exists an action choice under which there exists a safe partition, such that there exists a plan for all of its children.

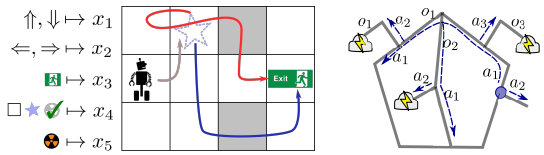
Let the number of actions and observations in $W$ be $|Y|$ and $|U|$, and the number of vertices be $|V|$. There are $2^{|U||V|}$ action choices to consider, in the worst case, for all the

world states in $\mathcal{W}$. The total number of partitions is a Bell number $B_{|U|}$, where $B_{n+1} = \sum_{k=0}^{n} C_n^k B_k$ and $B_0 = 1$. For each partition, the number of groups we must consider is $|U|$. To expand an action vertex in the search tree, the computation complexity is $2^{|U||V|}|U|B_{|U|}$. Similarly, the complexity to expand an observation vertex is $|Y|B_{|Y|}$. If the depth of the tree is $d$, then the computational complexity is $O(2^{|U||V|^d})$.

## 6 Experimental results

We implemented all the algorithms in this paper, the mainly using Python. The problem SEEK$_{\boldsymbol{x}}$ was implemented with both the algorithm we propose and via specification in computation tree logic (CTL) (and then utilizing the `nuXmv` model-checker). All executions in this section used a OSX laptop with a 2.4 GHz Intel Core i5 processor.

To experiment we constructed a $3 \times 4$ grid for the nuclear inspection scenario of Fig. 1. Including the differing facility types and radioactivity status, the world graph is a p-graph with 96 vertices before state-determined expansion (154 vertices for the state-determined form). The robot can move left, right, up, down one block at a time. After the robot's movement, it receives 5 possible observations: pebble bed facility or not (only when located at the blue star), radioactivity high or low when located at one of the '**?**' cells, and cell is an exit. But the observer only knows the image of the actions and observations under a label map. The stipulation requires that the observer should learn the radioactivity strength, but should never know the facility type.




(a) (b)

Fig. 7: The scenario and results for SEEK$_{\boldsymbol{x}}$ and SEEK$_{\boldsymbol{x},\boldsymbol{\lambda}}$ problem: (a) shows the plan found in the nuclear inspection scenario, when the observer knows nothing about robot's plan (The robot traces the gray arrow, then the blue one if blue light is seen, the red one otherwise.) (b) shows the pentagonal world in SEEK$_{\boldsymbol{x},\boldsymbol{\lambda}}$, where the robot moves along the gray lines.

Firstly, we SEEK the plan in the nuclear inspection scenario with a label map shown in Fig. 7a. A plan can be found (with the world graph disclosed, $D = W$). It takes 11 seconds for the AND–OR search and 24 seconds for the CTL-based implementation to find their solutions. The CTL solver takes longer, but it prioritizes finding the plan of shortest length first. The plan found by CTL is shown in Fig. 7a. As the plan found by AND–OR search is lengthy, we omit it.

Since, for the nuclear inspection scenario, SEEK$_{\boldsymbol{x},\boldsymbol{\lambda}}$ doesn't return any result within reasonable time we opted to examine a smaller problem. Here a robot moves in the pentagonal world shown in Fig. 7b. The robot can either decide to loop in the world ($a_1$) or exit the loop at some point ($a_2$ or $a_3$). We wish to find a plan and label map pair so that the robot can reach some charging station. The observer should not be able

to distinguish the robot's position when at either of the top two charging locations. $\textsc{Seek}_{\boldsymbol{x},\boldsymbol{\lambda}}$ gives a plan which moves forward 6 times and then exits at the next time step. Additionally, to disguise the actions and observations after the exit, it maps $h(a_2) = h(a_3)$ and $h(o_1) = h(o_3)$. Note that in this problem, the robot reaches a goal, without considering the stipulations, by taking the exit at the next time step. The stipulations force the robot to navigate at least one loop in the world to conflate state for the sake of the observer.

## 7 Conclusion

This paper continues a line of work on planning with constraints imposed on knowledge- or belief-states. Our contribution is a substantial generalization of prior models, though, as we see in the section reporting experiments, with grim implications for computational requirements. Future work might consider techniques that incorporate costs, informed methods (with appropriate heuristics), and other ways to solve certain instances quickly.

## References

1. L. Vaas. (2017) Privacy dust-up as Roomba maker mulls selling maps of users' homes. [Online]. Available: https://nakedsecurity.sophos.com/2017/07/26/
2. J. M. O'Kane, "On the value of ignorance: Balancing tracking and privacy using a two-bit sensor," in *WAFR*, 2008, pp. 235–249.
3. J. M. O'Kane and D. A. Shell, "Automatic design of discreet discrete filters," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2015, pp. 353–360.
4. Y. Zhang and D. A. Shell, "Complete characterization of a class of privacy-preserving tracking problems," *Intern. J. of Robotics Research—in WAFR'16 special issue*, 2018.
5. L. Takayama, D. Dooley, and W. Ju, "Expressing Thought: Improving Robot Readability with Animation Principles," in *Proceedings of the International Conference on Human-Robot Interaction (HRI'11)*, Lausanne, Switzerland, Mar. 2011, pp. 69–76.
6. A. D. Dragan, "Robot Planning with Mathematical Models of Human State and Action," arXiv preprint arXiv:1705.04226, 2017.
7. R. A. Knepper, C. I. Mavrogiannis, J. Proft, and C. Liang, "Implicit Communication in a Joint Action," in *Proceedings of the International Conference on Human-Robot Interaction (HRI'17)*, Vienna, Austria, Mar. 2017, pp. 283–292.
8. S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.
9. P. Masters and S. Sardina, "Deceptive Path-Planning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Melbourne, Australia, Aug. 2017, pp. 4368–4375.
10. A. D. Dragan, R. Holladay, and S. S. Srinivasa, "Deceptive Robot Motion: Synthesis, Analysis and Experiments," *Autonomous Robots*, vol. 39, no. 3, pp. 331–345, Oct. 2015.
11. Y.-C. Wu, V. Raman, S. Lafortune, and S. A. Seshia, "Obfuscator synthesis for privacy and utility," in *NASA Formal Methods Symposium*. Springer, 2016, pp. 133–149.
12. F. Z. Saberifar, S. Ghasemlou, D. A. Shell, and J. M. O'Kane, "Toward a language-theoretic foundation for planning and filtering," *Intern. J. of Robotics Research—in WAFR'16 special issue*, 2018.
13. J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.