Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020)
Main Track

# Reasoning About Plan Robustness Versus Plan Cost for Partially Informed Agents

**Sarah Keren**[1,2], **Sara Bernardini**[3], **Kofi Kwapong**[1] and **David C. Parkes**[1]
[1]School of Engineering and Applied Sciences, Harvard University
[2]Center for Research on Computation and Society at Harvard University
[3]Department of Computer Science, Royal Holloway University of London
skeren@seas.harvard.edu, sara.bernardini@rhul.ac.uk, kwapongk@college.harvard.edu, parkes@eecs.harvard.edu

### Abstract

A common approach to planning with partial information is *replanning*: compute a plan based on assumptions about unknown information and replan if these assumptions are refuted during execution. To date, most planners with incomplete information have been designed to provide guarantees on completeness and soundness for the generated plans. Switching focus to performance, we measure the *robustness* of a plan, which quantifies the plan's ability to avoid failure. Given a plan and an agent's *belief*, which describes the set of states it deems as possible, robustness counts the number of world states in the belief from which the plan will achieve the goal without the need to replan. We formally describe the trade-off between robustness and plan cost and offer a solver that is guaranteed to produce plans that satisfy a required level of robustness. By evaluating our approach on a set of standard benchmarks, we demonstrate how it can improve the performance of a partially informed agent.

## 1 Introduction

In many realistic applications, agents need to form a plan for reaching a goal with only partial information about their surroundings. That is, an agent needs to decide how to act based on its current *belief*, which describes the set of states it deems as possible. Among the different approaches to planning with partial information (Bonet and Geffner 2000; Palacios and Geffner 2009; Bonet and Geffner 2011; Brafman and Shani 2014; Muise, Belle, and McIlraith 2014; Brenner and Nebel 2009; Albore, Palacios, and Geffner 2009; Maliah et al. 2014), we adopt the *replanning* approach of Bonet and Geffner (2011), where an agent computes a plan while making assumptions about the values of unknown variables. It then executes the generated plan until reaching the goal or an assumption is refuted, in which case it replans.

For planning, Bonet and Geffner (2011) suggest the $KP$ compilation, which transforms the planning problem into a corresponding classical planning formulation. The actions of the compiled problem are: (i) execution actions representing the physical behavior of the agent in the environment; (ii) the assumptions it can choose to make about the values of unknown variables; and (iii) the reasoning it can do to infer new information from the data it has collected.

A key merit of the $KP$ compilation lies in the ability to use any off-the-shelf classical planner to find a solution to the compiled problem. However, a major shortcoming is that, while it comes with completeness and soundness guarantees, it does not provide a way to reason about the agent's performance in regard to the different types of actions used by the solution to the compiled problem. For example, an optimal solution may be one that favors performing more physical actions over using acquired data to infer new information. This would not be appropriate in contexts in which plans are generated to control a robot whose physical actions have a high cost, but where the use of sensory information for decision making has a low cost. On the other hand, an optimal solution might involve making many assumptions and few physical actions, which, due to the increased chance of replanning, would not be ideal in settings such as underwater missions where access to a planner may only be possible via a limited communication channel.

We explore this trade-off and provide a new translation that can accommodate for various agent preferences regarding which performance criteria to prioritize. In particular, we assume that the cost of reasoning about available information is negligible but allow the agent to decide to what extent it wants to minimize the need for replanning. For this, we adopt a *robustness* measure that quantifies the plan's ability to avoid failures (Nguyen, Sreedharan, and Kambhampati 2017). Given a plan and an agent's belief, robustness counts the number of states in the belief from which the plan will achieve the goal without replanning.

Given our measure for plan robustness, we suggest the $KP_{rob}$ compilation, extending $KP$ by allowing the agent to specify its required level of robustness. We show that by tuning the cost associated with making assumptions, our compilation supports agents with different performance preferences. In particular, increasing the cost of assumptions promotes robust plans in which the need to replan is minimized in favor of performing more execution actions. On the flip side, reducing the cost of assumptions promotes plans that compromise robustness in order to minimize the execution cost of the generated plan.

**Example 1.** *Consider Figure 1, depicting a simple example from the Wumpus domain (Russell and Norvig 2016), where a partially informed agent aims to reach the cell marked by 'Goal' without encountering a deadly wumpus or pit. The agent, which enters the system at 'Start', always knows its current position and has a map of the environment (with the*

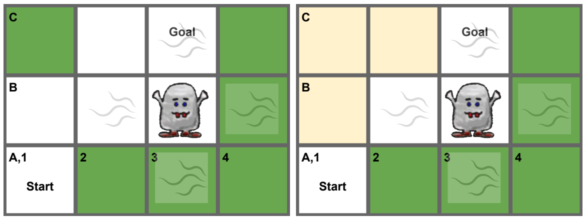


Figure 1: Wumpus Domain - Example 1

*location of the goal). It initially does not know the number and locations of deadly wumpuses and pits in the environment. When in a cell adjacent to a wumpus or pit, it senses a 'stench' or 'breeze', respectively. Unless the agent has previous information about adjacent cells, the agent does not know which direction the signal originates from. In our example, there is one wumpus in cell $(B, 3)$ and no pits.*

*We assume the agent is optimistic when planning but conservative when acting (Bonet and Geffner 2011). Optimistic planning means the agent can make assumptions about unknown variables in order to compute a plan. Conservative means it only takes actions for which the outcome is known; i.e. when the agent senses a stench or breeze, it only moves to an adjacent cell if it can infer it is 'safe', (i.e. with no wumpus or pit); it backtracks otherwise.*

*We consider two settings that differ in the initial information of the agent. In Figure 1[left], the agent knows that cells $(A, 2)$, $(A, 3)$, $(A, 4)$, $(B, 4)$, $(C, 1)$, $(C, 4)$ and the goal $(C, 3)$ are all safe. The optimal plan generated by the $KP$ compilation for the initial belief goes right three steps, up two steps and then left one step (6 physical movements and no assumptions or ramifications). The $KP_{rob}$ compilation can instead account for different agent preferences. If the agent wants to be maximally robust and make the minimal number of assumptions, its plan will be the same as for $KP$. The agent can also decide to execute a less robust plan that minimizes the number of physical actions. In this case, it will assume cells $(B, 1)$ and $(C, 2)$ have neither pits nor wumpuses and go up two steps and then right two steps.*

*In Figure 1[right], the agent knows that the goal cell $(C, 3)$, and cells $(A, 2)$, $(A, 3)$, $(A, 4)$, $(B, 4)$ and $(C, 4)$ are safe. It also knows that cells $(B, 1)$, $(C, 1)$ and $(C, 2)$ have neither pits nor wumpuses. This means it is possible to infer they are safe using ramification actions, but the plan generated by the $KP$ compilation is the same as for the left example. In contrast, by ignoring the cost of reasoning, the $KP_{rob}$ compilation finds a plan with fewer movements (for any specified robustness preference), which first infers cells $(B, 1)$, $(C, 1)$ and $(C, 2)$ are safe at no cost, and then follows the same execution actions as in the left case.*

Our contributions are threefold. First, we provide the first principled way to account for the trade-off between plan cost and robustness. Second, we offer the $KP_{rob}$ compilation as an extension of $KP$ that allows the user to specify its desired level of robustness, and we specify conditions under which optimal solutions found using $KP_{rob}$ are guaranteed to comply with the specified robustness. Finally, we compare our approach against the $KP$ compilation on a set of various robustness specifications that range from conservative agents that want to avoid replanning to agents that are willing to compromise robustness for efficiency of physical movements. Our evaluation on a set of standard benchmarks shows that plans with various levels of robustness can be acquired efficiently. We also demonstrate how increased robustness reduces the need to replan.

# 2 Preliminaries and Related Work

## 2.1 Planning under Partial Observability

We follow Bonet and Geffner's (2011) approach to modeling agents with partial knowledge and consider planning under partial observability defined as follows.

**Definition 1** (**Planning under partial observability.**) *A* planning under partial observability (PPO) problem *is a tuple $\mathcal{P} = \langle \mathcal{F}, \mathcal{A}, I, G, \mathcal{O} \rangle$ where $\mathcal{F}$ is a set of fluent symbols, $\mathcal{A}$ is a set of deterministic actions, $I$ is a set of* clauses *over $\mathcal{F}$-literals defining the initial situation, $G$ is a set of $\mathcal{F}$-literals defining the goal condition, and $\mathcal{O}$ represents the agent's sensor model.*

An action $a \in \mathcal{A}$ has a set of $\mathcal{F}$-literals preconditions, and a set of conditional effects $C \rightarrow L$, where $C$ is a set of $\mathcal{F}$-literals and $L$ is an $\mathcal{F}$-literal. The sensor model $\mathcal{O}$ is a set of observations $o \in \mathcal{O}$ represented as pairs $(C, L)$ where $C$ is a set of $\mathcal{F}$-literals and $L$ is a positive fluent, indicating that the value of $L$ is observable when $C$ is true. Each observation $o = (C, L)$ can be conceived as a sensor on the value of $L$ that is activated when $C$ is true.

A state $s$ is a truth valuation over the fluents $\mathcal{F}$ ('true' or 'false'). For an agent, the value of a fluent may be known or unknown. A fluent is *hidden* if its true value is unknown. A *belief* $b$ is a non-empty collection of states the agent deems as possible at some point. A formula $\mathbb{F}$ *holds* in $b$ if it holds for every state $s \in b$. An action $a$ is *applicable* in $b$ if the preconditions of $a$ hold in $b$, and the *successor* belief $b^{'}$ is the set of states that results from applying the action $a$ to each state $s$ in $b$. When an observation $o = (C, L)$ is activated, the successor belief is the set of states in $b$ that agree on $L$ (i.e., the set of states where fluent $L$ has the sensed value). The initial belief is the set of states that satisfy $I$, and the goal belief are those that satisfy $G$. A formula is *invariant* if it is true in each possible initial state and remains true in any state reachable from it. A *history* is a sequence of actions and beliefs $h = b_0, a_0, b_1, a_1, \ldots, b_n, a_n, b_{n+1}$, s.t. $a_i$ is applicable in $b_i$. Each action $a$ can be associated with a cost $\mathcal{C}(a)$, and the cost of history $h$, denoted $\mathcal{C}_a(h) = \Sigma_i \mathcal{C}(a_i)$, is the accumulated cost of the performed actions (equivalent to path length when action cost is uniform). A history is *complete* if the agent performing the actions reaches a goal belief. A solution to a PPO problem is a *policy* $\pi$, which is a partial mapping from beliefs to actions. Specifically, a solution can be a *plan* $\pi$ which is a sequence of actions.

## 2.2 Approaches to Online Planning

There are two main approaches to planning with partial information: *offline planning* and *online planning* (Brafman and Shani 2014). In the offline approach, a complete plan tree is generated to account for all the contingencies that may arise. This tree may grow exponentially in the number of problem variables, making it an impractical approach when computation time and resources are limited in all but simple problems. In the online approach, the agent makes local decisions on how to behave next, which can typically be generated more quickly but cannot provide the same guarantees as the offline approach.

A variety of both offline and online approaches have been developed to solve PPO planning (Bonet and Geffner 2000; Palacios and Geffner 2009; Bonet and Geffner 2011; Brafman and Shani 2014; Muise, Belle, and McIlraith 2014; Brenner and Nebel 2009; Albore, Palacios, and Geffner 2009; Maliah et al. 2014; Bonet and Geffner 2014b; Koenig, Tovey, and Smirnov 2003). A common technique for online planning is *replanning* (Zelinsky 1992). Here, the agent finds a plan for its current state based on some simplification of its planning problem and executes a prefix of the plan until discrepancies between the plan and the information acquired during the execution emerge and require replanning.

Most existing online approaches focus on providing soundness and completeness guarantees on the generated plans or on the task of *belief tracking*, which is the task of updating the belief with new information (Bonet and Geffner 2014a). Specifically, Bonet and Geffner (2011) show that the linear $KP$ translation, which we extend in this work, is sound and complete for problems with fully connected state spaces that are *simple*, i.e. the non-unary clauses in the initial belief are all invariant, and no hidden fluents appear in the body of a conditional effect. However, they do not consider the generated plan's ability to avoid the need to replan, which is instead our focus.

One exception is the recent work by Shmaryahu, Shani, and Hoffmann (2019), who provide various comparative criteria for plans and policies for PPO planning problems. One suggested criteria is *robustness to mishaps*, according to which unlikely outcomes are pruned when exploring the search tree. Similarly, Nguyen, Sreedharan and Kambhampati (2017) consider problems with incomplete specification of the domain models and use *robustness* to represent the probability mass of possible models under which a plan achieves the goals. They present two approaches to synthesizing plans that maximize robustness given the agent's current belief, while ignoring plan cost. We instead use the robustness measure to count the number of states in a belief in which the calculated plan can be executed and focus on providing plans that comply with a user specified level of robustness. We study the trade-off between the required robustness and the execution cost of the generated plan.

A special case of the $KP_{rob}$ compilation we present here was informally described in Keren et al. (2020) for partially informed agents that require plans that make the minimal number of assumptions among those that minimize the execution cost. Here, we offer the first formal definition of this approach, extending it to allow the user to specify its desired level of robustness, and specify conditions under which optimal solutions found using $KP_{rob}$ are guaranteed to comply with the specified robustness level.

## 2.3 The K-replanner and the $KP$ Translation

To represent our acting agents, we use the *K-replanner* by Bonet and Geffner (2011), which follows the replanning approach. The K-replanner constructs a policy that solves a PPO problem $P$ in an online fashion, setting the response to the current belief as the prefix of a plan obtained with an off-the-shelf classical planner.

**Algorithm 1** K-replanner ($PPO$ problem $\mathcal{P}$)

1: $b_{cur} = b_0$
2: **while** $G$ doesn't hold in $b_{cur}$ (while goal belief not achieved) **do**
3: $\pi_{cur} \leftarrow ComputePlan(\mathcal{P}, b_{cur})$
4: **if** $\mathcal{P}$ has no solution for $b_{cur}$ **then**
5: return { FAIL}
6: **end if**
7: **for** action $a$ in $\pi_{cur}$ **do**
8: **if** $a$ is not applicable in $b_{cur}$ **then**
9: break (return to Line 2)
10: **end if**
11: $b_{cur} \leftarrow Apply(a, \mathcal{P}, b_{cur})$ (apply $a$ to $b_{cur}$, fire the applicable sensors and update the current belief)
12: **end for**
13: **end while**
14: return { SUCCESS}

The pseudo code for the K-replanner is given in Algorithm 1. After initialization of the agent's belief (Line 1), the iterative planning process, which continues until the goal belief is reached, starts at Line 2. At each stage, a plan $\pi_{cur}$ is computed for the current belief (Line 3). If the current problem has no solution, the solver fails (Line 4). Otherwise, actions in the generated plan are applied in a loop (Line 7). If an action is not applicable (Line 8), replanning is performed by returning to Line 2. Otherwise, the action is applied and the belief is updated after firing (activating) applicable sensors and performing inference to deduce new information. Line 14 is reached if a goal belief is achieved, in which case 'SUCCESS' is returned.

Left undefined in Algorithm 1 is how to generate a plan for the current belief in Line 3. A practical approach is to produce a plan based on a partial, inaccurate or simplified version of the actual planning problem. Bonet and Geffner (2011) use the $KP$ translation[1] for this, which transforms the partially observable planning problem into a classical planning problem.

**Definition 2** ($KP$ **Translation).** *For a (PPO) problem* $P = \langle \mathcal{F}, \mathcal{A}, I, G, \mathcal{O} \rangle$, $KP = \langle \mathcal{F}', I', G', \mathcal{A}' \rangle$, *is the fully observable problem where*

- $\mathcal{F}' = \{KL, K\neg L : L \in \mathcal{F}\}$
- $I' = \{KL : L \in I\}$
- $G' = \{KL : L \in G\}$

[1]Note that Bonet and Geffner (2011) indicate their translation as $K(\mathcal{P})$. We renamed it as $KP$ because, as it will become apparent in Section 4, by following the original convention, we would have had to add another parameter (i.e. the cost of assumptions) to $K(\mathcal{P})$, making the name impractical to use.

- $\mathcal{A}' = \{\mathcal{A}'_{exe} \cup \mathcal{A}'_{as} \cup \mathcal{A}'_{ram}\}$ *where*
  - $\mathcal{A}'_{exe}$ *includes all actions* $a \in \mathcal{A}$*, but with each precondition* $L$ *replaced by* $KL$*, and each conditional effect* $C \rightarrow L$ *replaced by* $KC \rightarrow KL$ *and* $\neg K \neg C \rightarrow \neg K \neg L$*.*
  - $\mathcal{A}'_{as} = \{a_{(C,L)}, a_{(C,\neg L)} | o = (C, L) \in \mathcal{O}\}$ *where*
    - $prec(a_{(C,L)}) = \{KC, \neg KL, \neg K \neg L\}$ *and* $eff(a_{(C,L)}) = \{KL\})$
    - $prec(a_{(C,\neg L)}) = \{KC, \neg KL, \neg K \neg L\}$ *and* $eff(a_{(C,\neg L)}) = \{K \neg L\}$
  - $\mathcal{A}'_{ram} = \{a_{ram} |$ *for invariants* $\neg C \vee L$ *in* $I\}$ *where*
    - $prec(a_{ram}) = \{KC\}$ *and*
    - $eff(a_{ram}) = \{KL\}$

At the core of the $KP$ translation is the substitution of each fluent $L$ in the original problem with a pair of fluents $KL$ and $K \neg L$, representing whether $L$ is known to be true or false, respectively (Albore, Palacios, and Geffner 2009). The action set $\mathcal{A}' = \mathcal{A}'_{exe} \cup \mathcal{A}'_{as} \cup \mathcal{A}'_{ram}$ in the transformed problem contains three types of actions. The set $\mathcal{A}'_{exe}$ denotes the original set of actions (the *execution actions*): each original action $a \in \mathcal{A}$ is transformed into an equivalent action that replaces the use of every literal $L$ ($\neg L$), with its corresponding fluent $KL$ ($K \neg L$). The set $\mathcal{A}'_{as}$ represents assumptions that can be made about observations that may be collected during execution. Each observation $(C, L)$ is translated into two deterministic actions, one for each possible value of $L$. These actions allow the solver to compute a plan while choosing preferred values of (making assumptions about) unknown variables. The set $\mathcal{A}'_{ram}$, which we refer to as *ramification actions*, corresponds to the inference actions performed by the agent. Each invariant clause defined in $I$ is translated into a set of ramification actions that can be applied to set the truth value of relevant variables as new sensed information is collected from the environment.

This representation captures the underlying planning problem at the knowledge level (Petrick and Bacchus 2002), accounting for the exploratory behavior of a partially informed agent. In our example, an execution action $a_{exe} \in \mathcal{A}'_{exe}$ corresponds to a movement between adjacent cells. An action $a_{as} \in \mathcal{A}'_{as}$ can correspond to the agent assuming that a cell on its planned path has no pit. A ramification action $a_{ram} \in \mathcal{A}'_{ram}$ can be activated to infer that a cell is safe when an agent is in an adjacent cell and does not sense a breeze or stench.

By using the $KP$ transformation, the agent is following a *planning under optimism* approach. The agent plans while making the most convenient assumptions about the values of hidden variables. If during execution an observation refutes some assumptions, the agent revises its assumptions and re-plans accordingly. To demonstrate, consider an agent in the setting depicted in Figure 1[left] that follows a plan that includes moving to cell $(B, 2)$ on the way to the goal and is therefore based on the assumptions that the cell has no stench and no breeze. When the agent reaches cell $(B, 2)$, it senses the stench, and, since one of its assumptions is refuted, the agent needs to compute a new plan to the goal.

## 3 Assessing Plan Quality

Using the PPO problem formulated in Definition 1, an agent's belief represents the set of states deemed as possible at a given stage. To plan its behavior, the agent can make assumptions about the unknown variables by selecting their values. During execution, each time an action is performed, all applicable sensors are fired. Assumptions are refuted when a sensed value contradicts an assumption that was made, raising the need to replan.

We are interested in assessing and comparing the efficiency of plans produced by different planners. We consider a *plan* of a PPO problem $\mathcal{P}$ to be a sequence $\pi = a_0, a_1, \cdots, a_n$ of execution actions $a_i \in \mathcal{A}$ (excluding any ramification or assumptions). One natural performance measure by which to evaluate a plan is by the accumulated cost of its actions, denoted $\mathcal{C}_a(\pi) = \Sigma_i \mathcal{C}(a_i)$.

We also consider the *robustness* of a plan, representing the number of possible world states in which the plan is *executable*, meaning that the entire sequence of actions can be applied from that state without any need for replanning.

**Definition 3 (Executable Plan).** *Given a PPO problem* $\mathcal{P}$*, let* $S_{\mathcal{P}}$ *be the set of possible world states in* $\mathcal{P}$*. A plan* $\pi = \langle a_0, a_1, \ldots, a_n \rangle$ *is* executable *in* $s \in S_{\mathcal{P}}$ *if* $a_0$ *is applicable in* $s$ *and for any* $0 < i \leq n$*,* $a_i$ *is applicable in* $a_{i-1}(\ldots(a_0(s)))$*.*

In the context of PPO planning, a plan that an agent follows may not be executable in all the states in the initial belief $b_0$. In particular, it is not executable in the true state if an assumption that is made at planning time is refuted during execution. The *robustness* of a plan is defined with regard to the agent's current belief, counting the number of states in the belief in which the plan is executable.

**Definition 4 (Plan robustness).** *Given a PPO problem* $\mathcal{P}$*, the* robustness $R_{\mathcal{P}}(\pi, b)$ *of a plan* $\pi$ *w.r.t. a belief* $b$ *in* $\mathcal{P}$ *is the number of states* $s \in b$ *s.t.* $\pi$ *is executable in* $s$*.*

Considering Figure 1[left], a plan that goes up two steps and then right two steps has higher robustness than a plan that goes up, right, up, right since the latter relies on the extra assumptions that there is no stench and no breeze in cell $(B, 2)$.

Maximum robustness is achieved with a *conformant* plan (Palacios and Geffner 2009), which is a plan that does not rely on any assumptions and is executable for all the states in the initial belief. However, such a plan may not exist even when the goal can be achieved. Compromising robustness, an *optimistic* approach allows the planner to make assumptions about unknown variables, limiting the number of states in the belief in which the generated plan is executable.

We now define the *Full Observability Optimal* (FO-Optimal) cost of a PPO problem, which represents the minimum cost to goal that the agent would incur if it had full knowledge of the true world state.

**Definition 5 (FO-Optimal cost and plan).** *Given a PPO problem* $\mathcal{P}$*, the* Full Observability Optimal (FO-Optimal) cost *of* $\mathcal{P}$*, denoted by* $C^{FO^*}(\mathcal{P})$*, is the minimum cost of a plan from the initial true world state to a goal state. When there is no such plan,* $C^{FO^*}(\mathcal{P}) = \infty$*. Otherwise,* *a plan $\pi$ is* FO-Optimal *in $\mathcal{P}$ if it is executable in the initial true world state, it reaches a goal state, and its cost is $C^{FO^*}(\mathcal{P})$.*

Another cost that we consider is the *Optimistic-Optimal cost* of a PPO problem $\mathcal{P}$, which is the minimum cost among plans from the states in the initial belief to a goal belief.

**Definition 6 (Optimistic-Optimal cost and plan).** *Given a PPO problem $\mathcal{P}$, the* Optimistic-Optimal *cost of $\mathcal{P}$, denoted by $C^{OO}(\mathcal{P})$, is the minimum among the costs of plans that reach a goal belief from a state in the initial belief $b_0$. When there is no such plan, $C^{OO}(\mathcal{P}) = \infty$. Otherwise, a plan $\pi$ is an* Optimistic-Optimal *plan of $\mathcal{P}$ if it has the minimum cost among the plans that reach a goal belief from a state in the initial belief $b_0$.*

An Optimistic-Optimal plan is not necessarily executable for all states in the belief and will fail if it is not executable for the true world state. A useful observation is that the FO-Optimal cost serves as an upper bound on the Optimistic-Optimal cost.

**Lemma 1.** *Given a PPO problem $\mathcal{P}$ where the initial true world state is assumed to belong to $b_0$,*

$$C^{OO}(\mathcal{P}) \leq C^{FO^*}(\mathcal{P})$$

*Proof.* Making the correct assumptions for all unknown variables is a valid set of assumptions that yields a plan which costs $C^{FO^*}(\mathcal{P})$. □

A plan is *OP-Rob* if it has the highest robustness among the optimistic-optimal plans. A plan is *EXE-Rob* if it has the minimum cost among the plans to a goal belief that maximize robustness.

**Definition 7 (OP-Rob plan).** *Given a PPO problem $\mathcal{P}$, a plan $\pi$ is an* OP-Rob *plan for $\mathcal{P}$ if it is an Optimistic-Optimal plan of $\mathcal{P}$ and there is no other optimistic-optimal plan for $\mathcal{P}$ $\pi'$ s.t. $R_{\mathcal{P}}(\pi', b_0) > R_{\mathcal{P}}(\pi, b_0)$.*

**Definition 8 (EXE-Rob plan).** *Given a PPO problem $\mathcal{P}$, a plan $\pi$ is an* EXE-Rob *plan for $\mathcal{P}$ if, across all plans that reach a goal belief and maximize robustness, there is no plan with a lower cost.*

An OP-Rob plan succeeds without the need to replan for the maximum number of states in the initial belief among all optimistic-optimal plans. An EXE-Rob plan is a minimum cost plan among the plans that avoid replanning for the maximum number of states in the initial belief.

These two plan characteristics correspond to agents with different performance criteria. Agents such as large scale robots for which executing actions in the environment is extremely costly but replanning is cheap may prefer following OP-Rob plans. Agents for which (re)planning is costly may prefer following an EXE-Rob plan. This is relevant, for example, to robots performing an under-water mission in which planning is done by a controller above the surface with which communication is limited.

## 4 The $KP_{rob}$ Translation

The $KP$ translation by Bonet and Geffner (2011) (Definition 2) associates the same cost to all actions in the compiled planning problem, including assumption and ramification actions. As a consequence, there is no preference between applying any particular type of action in the compiled problem. As we demonstrate in Example 1, an optimal plan to $KP$ may be one that favors using a large number of execution actions rather than applying multiple ramifications or assumptions.

We seek a solver that can make assumptions, but will provide a user with the ability to define a desired trade-off between the execution cost of the generated plan and its robustness. In other words, we want a solver that supports different levels of deviation from plans with Optimistic-Optimal cost in favor of plans with increased robustness (which reduce the need for replanning).

For this, we assign a cost of 1 to execution actions $\mathcal{A}^{'}_{exe}$, which change the state of the world, and assign a cost of 0 to ramification actions $\mathcal{A}^{'}_{ram}$, which represent reasoning about available information. We assume that reasoning is done with no overheard and therefore do not consider it when computing the plan's cost. We assign a cost $\mathcal{C}^{''}_{as}$ to actions that represent making assumptions while planning. The user of the planner will make the choice of what this cost should be based on the required level of robustness.

**Definition 9 ($KP_{rob}$ translation).** *Given a PPO problem $P = \langle \mathcal{F}, \mathcal{A}, I, G, \mathcal{O} \rangle$ and a cost $\mathcal{C}^{''}_{as} \in \Re^{+} \cup \infty$ associated with making assumptions, the $KP_{rob}$* translation *is defined from $KP = \langle \mathcal{F}', I', G', \mathcal{A}' \rangle$ as the fully observable problem $P'' = \langle \mathcal{F}', I', G', \mathcal{A}', \mathcal{C}'' \rangle$ where*

$$\mathcal{C}''(a) = \begin{cases} 1 & \text{if } a \in \mathcal{A}^{'}_{exe} \\ \mathcal{C}^{''}_{as} & \text{if } a \in \mathcal{A}^{'}_{as} \\ 0 & \text{if } a \in \mathcal{A}^{'}_{ram} \end{cases}$$

Given a plan $\bar{\pi}$ that is a solution to the $KP_{rob}$ translation, we let $AS(\bar{\pi})$ and $EX(\bar{\pi})$ represent the sequence of assumption actions and execution actions in $\bar{\pi}$, respectively. We use $|AS(\bar{\pi})|$ and $|EX(\bar{\pi})|$ to represent the respective sizes of these two sequences. Since ramification actions $\mathcal{A}^{'}_{ram}$ have zero cost, the cost of $\bar{\pi}$ is $\mathcal{C}(\bar{\pi}) = |EX(\bar{\pi})| + \mathcal{C}^{''}_{as} \cdot |AS(\bar{\pi})|$.

**Example 2.** *To demonstrate the $KP_{rob}$ translation, consider Figure 2 [left], which is similar to the setting described in Example 1, except that cells here can be either 'free' (traversable) or 'occupied' (non-travesable). The need to replan occurs when an agent tries to move to an occupied cell that it previously assumed to be free.*

*The agent starts at the cell marked by 'Start' and its goal is the cell marked by 'Goal'. At the initial state, the agent knows all cells are 'free', except for the cells marked by '?', for which the agent does not know the true value. Figure 2 [right] depicts five possible plans ($P1$ to $P5$) the agent may choose to follow (for clarity, we depict only the prefixes of the plans, which all continue by moving up to the fifth row, and left to the goal when relevant).*

*Plan $P1$ represents an OP-Rob plan since it minimizes the cost to goal (its execution cost is equal to $C^{OO}(\mathcal{P})$),*

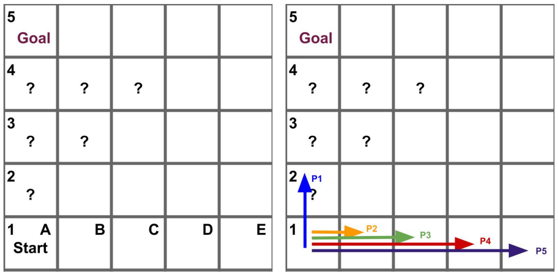


Figure 2: Demonstrating the $KP_{rob}$ translation

*while making the minimum number of assumptions (in this example, it is the only plan that minimizes execution cost). The solution to $KP_{rob}$ that corresponds to $P1$, called $\bar{\pi}_1$, includes 7 actions; 4 execution actions in $\mathcal{A}'_{exe}$ that move up to the goal and 3 assumption actions in $\mathcal{A}'_{as}$ (corresponding to the assumptions that cells $(A,2)$, $(A,3)$, and $(A,4)$ are 'free'). According to Definition 9, the cost of $\bar{\pi}_1$ is $\mathcal{C}(\bar{\pi}_1) = 3+4\cdot\mathcal{C}''_{as}$. Similarly, plan P4 represents an EXE-Rob plan since it is the shortest plan among the plans to the goal that minimizes the number of assumptions. The solution to $KP_{rob}$ that corresponds to $P4$, called $\bar{\pi}_4$, includes 10 execution actions in $\mathcal{A}'_{exe}$ and no assumption actions. The cost of $\bar{\pi}_4$ is $\mathcal{C}(\bar{\pi}_4) = 10$. As we will see below, the choice of which plan is returned by an optimal planner (that selects a plan that minimizes the total cost) can be controlled by setting the cost of assumptions $\mathcal{C}''_{as}$.*

## 4.1 Theoretical Analysis

This analysis has several objectives. First, we describe the relationship between assumptions that are part of a solution to the $KP_{rob}$ compilation and the robustness of the plan it represents. We then discuss the relationship between the cost $\mathcal{C}''_{as}$ associated with the assumptions and a guaranteed bound on the level of robustness of an optimal solution to $KP_{rob}$. Finally, we specify values to assign to $\mathcal{C}''_{as}$ in order to guarantee a specific level of robustness.

Let $|b|$ be the size of a belief $b$. Lemma 2 shows that for any solution $\bar{\pi}$ to the $KP_{rob}$ translation of a PPO problem a lower bound on its robustness is induced by the number of assumption actions in the solution.

**Lemma 2.** *Given a PPO problem $\mathcal{P}$, a cost $\mathcal{C}''_{as} \in \Re^+ \cup \infty$, and any solution $\bar{\pi}$ to $KP_{rob}$,*

$$R_{\mathcal{P}}(EX(\bar{\pi}), b_0) \geq \frac{|b_0|}{2^{|AS(\bar{\pi})|}}$$

*Proof.* Each assumption action in $\mathcal{A}'_{as}$ sets the value of an unknown Boolean variable in $\mathcal{F}$, so each assumption action $a \in AS(\bar{\pi})$ in the set of assumptions made in a solution $\bar{\pi}$ may remove half of the states in a given belief. □

Next, we focus on optimal solutions to $KP_{rob}$ and show that if $\mathcal{C}''_{as} > 0$, their robustness can be directly computed by the number of assumptions in the plan.

**Corollary 1.** *Given a PPO problem $\mathcal{P}$, a cost $\mathcal{C}''_{as} \in \Re^+ \cup \infty$, and an optimal solution $\bar{\pi}^*$ to $KP_{rob}$, if $\mathcal{C}''_{as} > 0$ then*

$$R_{\mathcal{P}}(EX(\bar{\pi}^*), b_0) = \frac{|b_0|}{2^{|AS(\bar{\pi}^*)|}}$$

*Proof.* From Lemma 2, we know that $R_{\mathcal{P}}(EX(\bar{\pi}^*), b_0) \geq \frac{|b_0|}{2^{|AS(\bar{\pi}^*)|}}$. Assume to the contrary that $R_{\mathcal{P}}(EX(\bar{\pi}^*), b_0) > \frac{|b_0|}{2^{|AS(\bar{\pi}^*)|}}$. This means that there exists at least one assumption action $a \in AS(\bar{\pi}^*)$ that can be removed while yielding the same execution cost but higher robustness. This in turn means that there is a different solution $\bar{\pi}$ for which $EX(\bar{\pi}) = EX(\bar{\pi}^*)$ and $|AS(\bar{\pi})| \leq |AS(\bar{\pi}^*)| - 1$. However, according to Definition 9 such a plan will have a lower cost than $\bar{\pi}^*$, contradicting our assumption that $\bar{\pi}^*$ is an optimal solution and concluding our proof. □

Corollary 1 provides a way to compute the robustness of an optimal solution to $KP_{rob}$. We now specify values for the cost of assumptions to achieve a desired level of robustness. For this purpose, we use a solution $\bar{\pi}_{opr}$ to $KP_{rob}$ that represents an OP-Rob solution to $\mathcal{P}$. According to Definition 7, this means that $EX(\bar{\pi}_{opr})$ is an OP-Rob plan to $\mathcal{P}$, for which the cost is $C^{OO}(\mathcal{P})$. For now, we assume that such a solution is available. Later, in Theorem 2, we show how to compute it.

**Lemma 3.** *Given a PPO problem $\mathcal{P}$, a cost $\mathcal{C}''_{as} \in \Re^+ \cup \infty$, and an optimal solution $\bar{\pi}^*$ to $KP_{rob}$,*

$$|EX(\bar{\pi}^*)| - |EX(\bar{\pi}_{opr})| \leq \mathcal{C}''_{as} \cdot (|AS(\bar{\pi}_{opr})| - |AS(\bar{\pi}^*)|)$$

*where $\bar{\pi}_{opr}$ is a solution to $KP_{rob}$ that corresponds to an OP-Rob plan of $\mathcal{P}$.*

*Proof.* According to Definition 9, for any PPO problem $\mathcal{P}$, the cost of any solution $\bar{\pi}$ to $KP_{rob}$ is $|EX(\bar{\pi})| + \mathcal{C}''_{as} \cdot |AS(\bar{\pi})|$, which consists of the cost $|EX(\bar{\pi})|$ of its execution actions (equivalent to the number of actions since actions in $\mathcal{A}'_{exe}$ cost 1) and the cost of its assumption actions $AS(\bar{\pi})$, which cost $\mathcal{C}''_{as}$ each (ramification action have no cost).

Assume to the contrary that

$$|EX(\bar{\pi}^*)| - |EX(\bar{\pi}_{opr})| > \mathcal{C}''_{as} \cdot (|AS(\bar{\pi}_{opr})| - |AS(\bar{\pi}^*)|)$$

Therefore,

$$|EX(\bar{\pi}^*)| + \mathcal{C}''_{as} \cdot |AS(\bar{\pi}^*)| > |EX(\bar{\pi}_{opr})| + \mathcal{C}''_{as} \cdot (|AS(\bar{\pi}_{opr})|$$

This means that $\bar{\pi}_{opr}$ is a solution to $KP_{rob}$ with a lower cost than $\bar{\pi}^*$, thus contradicting our assumption that $\bar{\pi}^*$ is an optimal solution to $KP_{rob}$. □

Lemma 2 and Corollary 1 allow computing a lower bound on the robustness of any solution to $KP_{rob}$ and the exact robustness of an optimal solution given a cost associated with assumptions, respectively. Lemma 3 shows the maximum bound on the diversion from the minimum execution cost $C^{OO}(\mathcal{P})$ for a given cost $\mathcal{C}''_{as}$ assigned to assumptions.

To show how $KP_{rob}$ can be configured to account for various robustness requirements, we prove that the diversion from $C^{OO}(\mathcal{P})$ of an optimal solution $\bar{\pi}^*$ to $KP_{rob}$ is bounded by the maximum total cost of all possible assumptions.

**Theorem 1.** *Given a PPO problem $\mathcal{P}$, a cost $\mathcal{C}^{''}_{as} \in \Re^{+} \cup \infty$, and an optimal solution $\bar{\pi}^*$ to $KP_{rob}$,*

$$0 \leq |EX(\bar{\pi}^*)| - |EX(\bar{\pi}_{opr})| \leq \mathcal{C}^{''}_{as} \cdot \frac{|\mathcal{A}^{'}_{as}|}{2}$$

*where $\bar{\pi}_{opr}$ is a solution to $KP_{rob}$ that corresponds to an OP-Rob plan of $\mathcal{P}$.*

*Proof.* The fact that $0 \leq |EX(\bar{\pi}^*)| - |EX(\bar{\pi}_{opr})|$ is a direct consequence of Lemma 1, which shows that there is no plan with a cost lower than $|EX(\bar{\pi}_{opr})|$. For the other inequality, assume by contradiction that

$$|EX(\bar{\pi}^*)| - |EX(\bar{\pi}_{opr})| > \mathcal{C}^{''}_{as} \cdot \frac{|\mathcal{A}^{'}_{as}|}{2}$$

Since the set of assumptions $\mathcal{A}^{'}_{as}$ contains an action for each possible value of the fluents in $\mathcal{P}$ and only one value can be assigned to each variable, the maximum number of assumptions that can be made in a plan is $\frac{|\mathcal{A}^{'}_{as}|}{2}$. If the maximal number of assumptions are applied, at least one Optimistic-Optimal plan, for which the cost is $|EX(\bar{\pi}_{opr})|$, is executable. Therefore, any plan $\bar{\pi}$ for which $|EX(\bar{\pi})| - |EX(\bar{\pi}_{opr})|$ is larger than $\mathcal{C}^{''}_{as} \cdot \frac{|\mathcal{A}^{'}_{as}|}{2}$ will not be an optimal solution to $KP_{rob}$, thus contradicting our assumption. □

Theorem 1 shows the relationship between the divergence from the minimum cost of the generated plan and the cost associated with making assumptions. Specifically, it allows the user to set the cost of assumptions in a way that guarantees a specified bound on the allowed diversion from Optimistic-Optimal cost. At one extreme, by setting $\mathcal{C}^{''}_{as}$ and the allowed diversion from Optimistic-Optimal cost to zero, $KP_{rob}$ supports agents that want to follow minimum cost plans and are indifferent to robustness. At the other extreme, by excluding assumptions from the model, equivalent to assigning infinite cost to $\mathcal{C}^{''}_{as}$, $KP_{rob}$ supports conformant agents that are only willing to follow plans that are guaranteed to succeed. $KP_{rob}$ also supports any intermediate requirement between these two extremes.

We focus on two interesting special cases in this range and specify conditions under which $KP_{rob}$ is guaranteed to produce OP-Rob and EXE-Rob plans.

**Theorem 2.** *Given a PPO problem $\mathcal{P}$, a cost $\mathcal{C}^{''}_{as} \in \Re^{+} \cup \infty$, and an optimal solution $\bar{\pi}^*$ to $KP_{rob}$, if $0 < \mathcal{C}^{''}_{as} < \frac{2}{|\mathcal{A}^{'}_{as}|}$, then $EX(\bar{\pi}^*)$ is an OP-Rob plan for $\mathcal{P}$.*

*Proof.* Recall from the proof to Theorem 1 that the maximum number of assumptions that can be made in a plan is $\frac{|\mathcal{A}^{'}_{as}|}{2}$. Since $0 \leq \mathcal{C}^{''}_{as} < \frac{2}{|\mathcal{A}^{'}_{as}|}$, for any plan $\bar{\pi}$ of $KP_{rob}$,

$$0 \leq \mathcal{C}^{''}_{as} \cdot |AS(\bar{\pi})| \leq \mathcal{C}^{''}_{as} \cdot \frac{|\mathcal{A}^{'}_{as}|}{2} < \frac{|\mathcal{A}^{'}_{as}|}{2} \frac{2}{|\mathcal{A}^{'}_{as}|} = 1 \quad (1)$$

i.e. the maximum total cost of assumptions is strictly smaller than 1.

Turning now to the statement of the theorem, assume by contradiction that $EX(\bar{\pi}^*)$ is not an OP-Rob plan for $\mathcal{P}$. According to Definition 7, this means that either (1) there is a plan that achieves the goal with a smaller execution cost; or (2) there is a plan with the same execution cost that requires making fewer assumptions. We will consider these two cases separately.

If condition (1) holds, then there exists a plan $\bar{\pi}$ s.t. $|EX(\bar{\pi})| < |EX(\bar{\pi}^*)|$. Since $\bar{\pi}^*$ is an optimal solution to $KP_{rob}$, we know that

$$|EX(\bar{\pi}^*)| + \mathcal{C}^{''}_{as} \cdot |AS(\bar{\pi}^*)| \leq |EX(\bar{\pi})| + \mathcal{C}^{''}_{as} \cdot |AS(\bar{\pi})|$$

and therefore

$$|EX(\bar{\pi}^*)| - |EX(\bar{\pi})| \leq \mathcal{C}^{''}_{as} \cdot (|AS(\bar{\pi})| - |AS(\bar{\pi}^*)|)$$

Given our assumption that $|EX(\bar{\pi})| < |EX(\bar{\pi}^*)|$ and knowing that the minimum difference is at least 1 (the cost of a single execution action), we have that

$$1 < |EX(\bar{\pi}^*)| - |EX(\bar{\pi})| \leq \mathcal{C}^{''}_{as} \cdot (|AS(\bar{\pi})| - |AS(\bar{\pi}^*)|)$$

Specifically, $1 < \mathcal{C}^{''}_{as} \cdot (|AS(\bar{\pi})| - |AS(\bar{\pi}^*)|)$. Since the maximum number of assumptions is $\frac{|\mathcal{A}^{'}_{as}|}{2}$, we have that $|AS(\bar{\pi})| - |AS(\bar{\pi}^*)| \leq \frac{|\mathcal{A}^{'}_{as}|}{2}$ and therefore

$$1 < \mathcal{C}^{''}_{as} \cdot (|AS(\bar{\pi})| - |AS(\bar{\pi}^*)|) \leq \mathcal{C}^{''}_{as} \cdot \frac{|\mathcal{A}^{'}_{as}|}{2}$$

However, from Equation (1), we know that $\mathcal{C}^{''}_{as} \cdot \frac{|\mathcal{A}^{'}_{as}|}{2} < 1$, hence reaching the contradiction $1 < \mathcal{C}^{''}_{as} \cdot \frac{|\mathcal{A}^{'}_{as}|}{2} < 1$.

If condition (2) holds, then there exists a plan $\bar{\pi}$ s.t. $|EX(\bar{\pi})| = |EX(\bar{\pi}^*)|$ and $|AS(\bar{\pi})| < |AS(\bar{\pi}^*)|$. However, since $\bar{\pi}^*$ is an optimal solution to $KP_{rob}$, we know that

$$|EX(\bar{\pi}^*)| + \mathcal{C}^{''}_{as} \cdot |AS(\bar{\pi}^*)| \leq |EX(\bar{\pi})| + \mathcal{C}^{''}_{as} \cdot |AS(\bar{\pi})|$$

Since $|EX(\bar{\pi})| = |EX(\bar{\pi}^*)|$, this means that

$$\mathcal{C}^{''}_{as} \cdot |AS(\bar{\pi}^*)| \leq \mathcal{C}^{''}_{as} \cdot |AS(\bar{\pi})|,$$

and therefore, $|AS(\bar{\pi}^*)| \leq |AS(\bar{\pi})|$, which contradicts the assumption, and concludes the proof. □

Theorem 2 shows that an OP-Rob plan can be obtained by setting a small cost to each assumption. An OP-Rob plan is useful to compute the robustness bound of a solution to a $KP_{rob}$ translation of a PPO planning problem, as specified in Lemma 3. It is also useful in a variety of applications where acting in the real world is expensive but replanning can be done efficiently.

To conclude the theoretical analysis, we show how an EXE-Rob plan can be acquired by setting the cost of an assumption to be higher than the maximum execution cost of a plan. This is relevant in applications where failure and replanning are costly.

**Theorem 3.** *Given a PPO problem $\mathcal{P}$, a cost $\mathcal{C}^{''}_{as} \in \Re^{+} \cup \infty$, and an optimal solution $\bar{\pi}^*$ to $KP_{rob}$, if $|\mathcal{A}^{'}_{exe}| < \mathcal{C}^{''}_{as} < \infty$, then $EX(\bar{\pi}^*)$ is an EXE-Rob plan for $\mathcal{P}$.*

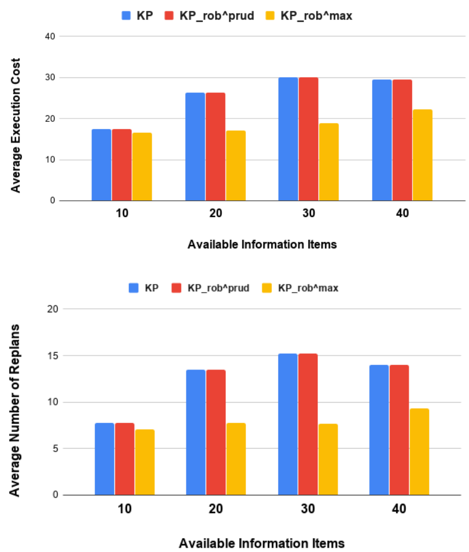


Figure 3: Results for C-BALLS

*Proof.* Assume to the contrary that $EX(\bar{\pi}^*)$ is not an EXE-Rob plan for $\mathcal{P}$. According to Definition 8, this means that there exists a solution $\bar{\pi}$ to $KP_{rob}$ that either (1) includes fewer assumptions than $\bar{\pi}^*$ or (2) makes the same number of assumptions but includes fewer execution actions. The first case is not possible since a single assumption costs more than applying the entire set of execution actions and the cost of $\bar{\pi}$ is strictly smaller than that of $\bar{\pi}^*$, contradicting our assumption that $\bar{\pi}^*$ is an optimal solution.

In the second case, $|EX(\bar{\pi}^*)| = |EX(\bar{\pi})|$ and $|AS(\bar{\pi})| < |AS(\bar{\pi}^*)|$. However, since $\bar{\pi}^*$ is an optimal solution to $KP_{rob}$,

$$|EX(\bar{\pi}^*)| + \mathcal{C}_{as}'' \cdot |AS(\bar{\pi}^*)| \leq |EX(\bar{\pi}^*)| + \mathcal{C}_{as}'' \cdot |AS(\bar{\pi})|,$$

and therefore $|AS(\bar{\pi}^*)| \leq |AS(\bar{\pi})|$, thus reaching a contradiction and concluding our proof. □

From a practical perspective, this theory provides a principled way to understand the implications of specific robustness requirements. For example, our approach can be used to compute in real-time the cost difference between an EXE-Rob and an OP-Rob plan, to understand the minimum robustness associated with each option, and to decide how to act based on available resources.

**Example 2** (continued)**.** *Returning to Example 2, the total number of assumptions that can be made is $2^6$ (each unknown value can be assumed to have one of two values). If the cost of assumptions is set to be less than $2^{-5}$, the optimal solution to $KP_{rob}$ will represent $P1$, the only OP-Rob plan in the example. Since this plan requires making 3 assumptions, the robustness of $P1$ is $2^3$. Similarly, if the cost of assumptions is set to be higher than $|\mathcal{A}_{exe}'|$, plan $P4$, which is the minimum cost plan among the plans to the goal that minimize the number of assumptions, is selected by an optimal solver. Since $P4$ does not include any assumptions, its robustness is $2^6$. The agent can compute both plans (and any intermediate plan) and decide how to act based on the differences between their execution costs and robustness.*

| | $KP$ | | $KP_{rob}^{prud}$ | | $KP_{rob}^{max}$ | | $KP_{rob}^{conf}$ | |
|---|---|---|---|---|---|---|---|---|
| | solved | time | solved | time | solved | time | solved | time |
| **WUMPUS** | 0.9 | 50.14 | 0.9 | 399.29 | 0.9 | 404.25 | 0.9 | 0.76 |
| **WUMPUS-KEY** | 0.75 | 65.8 | 0.75 | 542/3 | 0.75 | 601.2 | 0.75 | 1.65 |
| **UNIX** | 1.0 | 2.32 | 1.0 | 2.36 | 1.0 | 2.36 | 1.0 | 0.49 |
| **ROCKSAMPLE** | 1.0 | 135.70 | 1.0 | 316.22 | 1.0 | 312.10 | 1.0 | 0.27 |
| **LOGISTICS** | 0.8 | 231.58 | 0.8 | 218.22 | 0.8 | 222.06 | 0.8 | 49.12 |
| **C-BALLS** | 1.0 | 22.85 | 1.0 | 28.43 | 1.0 | 25.99 | 1.0 | 0.51 |
| **TRAIL** | 1.0 | 52.07 | 1.0 | 52.27 | 1.0 | 54.12 | 1.0 | 25.97 |

Table 1: Completed problems and computation time (in seconds)

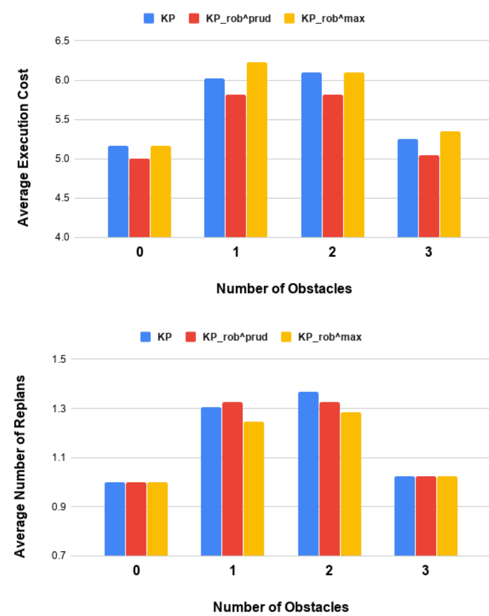


Figure 4: Results for WUMPUS

## 5 Empirical Evaluation

The empirical evaluation of the proposed method focuses on two objectives. First, having formally shown that the $KP_{rob}$ compilation is guaranteed to produce plans that comply with the robustness requirement set by the agent, we measure the computation time overhead of satisfying these guarantees. We then measure the total execution cost associated with each robustness level. We make use of seven PPO planning domains, adapted from Bonet and Geffner (2011) and Albore et al. (2009), and used by (Keren et al. 2020) [2].

- WUMPUS: the setting described in Example 1.
- WUMPUS-KEY: WUMPUS extended with keys, which need to be collected to achieve the goal destination. The initial position of the keys is not known.
- COLOR-BALLS: the agent navigates a grid to deliver balls of different and initially unknown colors to their per-color destinations. The agent can sense the color of the balls in its current cell.

[2]Our benchmark set, code, and results can be found in https://github.com/sarah-keren/krob-kr-2020

- TRAIL: the agent must follow a trail to reach a destination, while sensing the reachable cells surrounding it.
- LOGISTICS: trucks transport packages to their destinations. Sensors reveal the packages in a location.
- LOGISTICS-I: same as LOGISTICS but with additional dependencies regarding package locations (e.g., either package A or package B are in location 1).
- UNIX: a user wants to transfer a file to a specific location without knowing its initial location in a folder tree.
- FREESPACE: a robot navigating in an environment using an occupancy grid in order to reach some destination cell. It can sense walls from adjacent cells.

## 5.1 Computation Time

**Setup.** We use the K-replanner (Bonet and Geffner 2011) as the agent's solver, using Fast-Downward (Helmert 2006) with $A^*$ and the $h_{max}$ heuristic (caching precomputed values) to compute optimal plans. We used optimal solutions in order to provide the robustness guarantees we specify above. We compare $KP$ (Bonet and Geffner 2011) with 3 variations of $KP_{rob}$ discussed in Section 4: $KP_{rob}^{prud}$ (Keren et al. 2020) that finds OP-Rob plans (Definition 7), $KP_{rob}^{max}$ that finds EXE-Rob plans (Definition 8), and $KP_{rob}^{conf}$ that finds conformant plans (with no assumptions). All evaluations were performed on an Intel "Cascade Lake" machine with 64 GB memory allocated per domain and a 15 minutes time limit. To compare running times, we use 80 instances of each domain, where instances vary both in their initial state and initial belief.

**Results.** Table 1 compares the ratio of instances solved (*solved*) by each approach, and the average total running time (including planning and execution) in seconds (*time*) for instances solved by all approaches. The running time of both the $KP_{rob}^{prud}$ and $KP_{rob}^{max}$ approaches is higher than the $KP$ and $KP_{rob}^{conf}$ approaches, but the maximal running time difference (for WUMPUS and WUMPUS-KEY) is only about 8 times higher, and similar in all other domains. Moreover, all approaches achieve the same ratio of completed problems, showing that guarantees on plan robustness can be acquired efficiently. Consequently, the different compilations can be used as a pre-processing stage, allowing the user to compute the diversion from Optimistic-Optimal cost incurred for performing plans with higher robustness. Specifically, conformant solutions can be acquired quickly to reveal the cost a plan with guaranteed success.

## 5.2 Robustness and Execution Cost

**Setup.** To appreciate the effect robustness has on performance, we use the same empirical setup as above, but instead of a single setup for each initial state and belief, we randomly sample 100 initial states for each initial belief (e.g., distribution of pits in the WUMPUS domain). We use samples to represent states in the initial beliefs, since the state space is too big to enumerate exhaustively. For each setup we measure the total execution cost (physical actions taken) and the number of replans per execution.

**Results.** Figures 3 (top) and 4 (top) compare the execution costs for each approach for an increasing ratio of variables added to the agent's initial knowledge for C-BALLS and the number of obstacles for WUMPUS, respectively (excluding $KP_{rob}^{conf}$ since it failed for most instances). The results for C-BALLS show that for all cases, $KP_{rob}^{max}$'s increased robustness leads to a reduction in execution costs. In contrast, for WUMPUS, $KP_{rob}^{prud}$ achieved the lowest costs. To investigate this trend, we compare the number of replans for each approach (Figures 3 (bottom) and 4 (bottom)). The results for C-BALLS indicate that while $KP_{rob}^{max}$ uses the available information to generate plans that may have a higher cost than the Optimistic-Optimal plans followed by $KP$ and $KP_{rob}^{prud}$, it avoids the need to replan on more instances, which explains the reduced overall execution cost. Another interesting observation for C-BALLS is that execution cost might increase by giving more information to the agent (and reducing the size of its belief). This may happen if the added information misleads the agent to follow a plan that fails. While it is beyond the scope of this paper, it is an interesting trend to investigate.

# 6 Conclusion

We have presented a new approach to online planning for PPO problems. As an extension of Bonet and Geffner's (2011) $KP$ compilation, the $KP_{rob}$ compilation can be used to produce solutions that provide guarantees in regard to a user-specified level of robustness, a measure that we use to quantify a plan's ability to avoid failure. We evaluate our approach on a set of standard benchmarks and show that $KP_{rob}$ plans can be computed efficiently and reduce execution cost by reducing the need for replanning.

We have used robotic domains as motivation, but the $KP_{rob}$ approach is relevant to a variety of real-world applications where different users may have different performance criteria and can benefit from the ability to trade-off between execution cost and robustness. Such applications include navigation applications, where $KP_{rob}$ offers the user the option to specify to what extent she is willing to follow a longer but more reliable route to her destination. Another example is a supply-chain application that can allow a decision maker to quickly determine the benefit obtained by adopting a costlier but less risky fulfillment plan.

Across various applications, the planning space can be extremely complex, making it necessary to use abstract and simplified representations of the underlying planning problem in order to quickly compute solutions. In particular, non-deterministic sensor models such as those adopted here can provide a simplified representation of probabilistic sensor models, allowing efficient computation in that setting.

It would also be of interest to extend the present approach to accommodate probabilistic inference and robust planning in the setting of Partially Observable Markov Decision Processes (POMDPs) (Kaelbling, Littman, and Cassandra 1998). This will require extending the robustness measure to account for the probability that plans will succeed and the generation of robust plans that are most likely to achieve the goal.

## Acknowledgements

The authors thank Barbara J. Grosz, Hector Palacios, Sarath Sreedharan, Guy Shani, Blai Bonet and Hector Geffner for their helpful comments and suggestions. This work was supported by the Center for Research on Computation and Society at Harvard University and by the EPSRC Grant EP/S016473/1.

## References

Albore, A.; Palacios, H.; and Geffner, H. 2009. A translation-based approach to contingent planning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Bonet, B., and Geffner, H. 2000. Planning with incomplete information as heuristic search in belief space. In *Proceedings of the International Conference on Artificial Intelligence Planning Systems (AIPS)*.

Bonet, B., and Geffner, H. 2011. Planning under partial observability by classical replanning: Theory and experiments. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Bonet, B., and Geffner, H. 2014a. Belief tracking for planning with sensing: Width, complexity and approximations. *Journal of Artificial Intelligence Research (JAIR)* 50:923–970.

Bonet, B., and Geffner, H. 2014b. Flexible and scalable partially observable planning with linear translations. In *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Brafman, R., and Shani, G. 2014. Replanning in domains with partial information and sensing actions. *Journal of Artificial Intelligence Research (JAIR)* 45:565–600.

Brenner, M., and Nebel, B. 2009. Continual planning and acting in dynamic multiagent environments. *Autonomous Agents and Multi-Agent Systems* 19(3):297–331.

Helmert, M. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research (JAIR)* 26:191–246.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1-2):99–134.

Keren, S.; Xu, H.; Kwapong, K.; Parkes, D.; and Grosz, B. 2020. Information shaping for enhanced goal recognition of partially-informed agents. In *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Koenig, S.; Tovey, C.; and Smirnov, Y. 2003. Performance bounds for planning in unknown terrain. *Artificial Intelligence* 147(1-2):253–279.

Maliah, S.; Brafman, R.; Karpas, E.; and Shani, G. 2014. Partially observable online contingent planning using landmark heuristics. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*.

Muise, C.; Belle, V.; and McIlraith, S. A. 2014. Computing contingent plans via fully observable non-deterministic planning. In *Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Nguyen, T.; Sreedharan, S.; and Kambhampati, S. 2017. Robust planning with incomplete domain models. *Artificial Intelligence* 245:134–161.

Palacios, H., and Geffner, H. 2009. Compiling uncertainty away in conformant planning problems with bounded width. *Journal of Artificial Intelligence Research (JAIR)* 35:623–675.

Petrick, R. P., and Bacchus, F. 2002. A knowledge-based approach to planning with incomplete information and sensing. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*.

Russell, S. J., and Norvig, P. 2016. *Artificial intelligence: a modern approach*. Pearson Education.

Shmaryahu, D.; Shani, G.; and Hoffmann, J. 2019. Comparative criteria for partially observable contingent planning. *Autonomous Agents and Multi-Agent Systems* 33(5):481–517.

Zelinsky, A. 1992. A mobile robot exploration algorithm. *IEEE transactions on Robotics and Automation* 8(6):707–717.