Heuristic Search in Dual Space for Constrained Fixed-Horizon POMDPs with Durative Actions

Majid Khonji, Duoaa Khalifa*

Khalifa University
Abu Dhabi, UAE

Abstract

The Partially Observable Markov Decision Process (POMDP) is widely used in probabilistic planning for stochastic domains. However, current extensions, such as constrained and chance-constrained POMDPs, have limitations in modeling real-world planning problems because they assume that all actions have a fixed duration. To address this issue, we propose a unified model that encompasses durative POMDP and its constrained extensions. To solve the durative POMDP and its constrained extensions, we first convert them into an Integer Linear Programming (ILP) formulation. This approach leverages existing solvers in the ILP literature and provides a foundation for solving these problems. We then introduce a heuristic search approach that prunes the search space, which is guided by solving successive partial ILP programs. Our empirical evaluation results show that our approach outperforms the current state-of-the-art fixed-horizon chance-constrained POMDP solver.

1 Introduction

A crucial aspect of intelligent agents is their capacity to make well-informed decisions in uncertain environments. The Partially Observable Markov Decision Process (POMDP) is a mathematical model that enables agents to devise contingency plans aimed at maximizing expected utility (Sondik 1971; Kaelbling, Littman, and Cassandra 1998). This model has broad applications across various fields such as machine learning (Kaelbling, Littman, and Moore 1996), robotics (Spaan and Spaan 2004), healthcare (Hoey et al. 2014), predictive maintenance (Cassandra 1998), and more. POMDPs expand upon the well-studied Markov Decision Process (MDP) model by addressing situations where agents are unable to accurately observe the underlying state of the environment (Puterman 2014). Despite the expressive power, real-world applications often need to make several assumptions to fit the mathematical model.

Constrained POMDPs. A limitation of the standard POMDP is its focus on optimizing a single performance measure. In many domains, an agent might need to optimize an objective function while maintaining specific parameters within certain limits. For instance, a surveillance robot aiming to maximize search coverage may also need to keep its power consumption below battery capacity. As demonstrated in (Undurti and How 2010), modeling parameter violations as negative reward signals can lead to excessively risk-taking or risk-averse situations. Additionally, manually tuning reward signals places extra burden on the algorithm designer, and it can be challenging to thoroughly test all possible scenarios.

The Constrained POMDP (C-POMDP) is a mathematical model that addresses multi-criteria requirements by optimizing one criterion while bounding the others (Isom, Meyn, and Braatz 2008). In risk-sensitive applications, a practical approach is to limit the probability of failure rather than the expected cost value. For example, a planetary rover should constrain the *probability* of adverse events such as navigating near a cliff or colliding with obstacles (Mausam et al. 2005). This extension is often referred to as chance-Constrained POMDP (CC-POMDP) (Santana, Thiébaux, and Williams 2016).

Solution Methods for Constrained POMDPs. The focus in the literature has been primarily on the fully observable case, i.e., the *constrained* MDP (C-MDP). Comprehensive theoretical analysis is presented in (Altman 1999; Feinberg and Shwartz 1996). Several efficient algorithms are also proposed in the literature (e.g., see (Ono, Kuwata, and Balaram 2012; Trevizan et al. 2016; Hong et al. 2021; Alyassi and Khonji 2021)). The partially observable case is more recent (Isom, Meyn, and Braatz 2008), and the research is also less mature. Current methods span from extensions of dynamic programming (Isom, Meyn, and Braatz 2008), point-based value iteration (Kim et al. 2011), approximate linear programming (LP) (Poupart et al. 2015), and more recently, a heuristic forward search approach for CC-POMDP (Santana, Thiébaux, and Williams 2016). These techniques either compromise on the optimality (where policies might arbitrarily deviate from the optimal as in (Santana, Thiébaux, and Williams 2016)), or violate feasibility, as in the approximate methods (e.g., (Poupart et al. 2015)). In this work, we follow an integer linear programming (ILP) approach emphasizing risk-sensitive POMDP settings. Current LP-based approaches consider only stochastic policies (Poupart

*This work was supported by Khalifa University under Award Ref. CIRA-2020-286, KKJRC-2019-Trans1, and KUCARS. Majid Khonji and Duoaa Khalifa are with EECS Department (emails: {majid.khonji, duoaa.khalifa}@ku.ac.ae).

et al. 2015), some of which return optimal solutions but arguably not suitable for risk-sensitive applications (Santana, Thiébaux, and Williams 2016; Dolgov and Durfee 2005). A stochastic policy for C-POMDPs satisfies a constraint on expectation, while in CC-POMDPs, constraints should be satisfied on the percentile. This work focuses on deterministic policies that respect the percentile constraint; the stochastic version remains open for future research.

Durative Actions. Another important shortcoming of POMDP is that actions are assumed to have a fixed duration. Such an assumption is unnatural in many application domains and often requires modeling artifacts such as discretization, leading to larger and inefficient models. Semi-Markov decision process (SMDP) is a well-studied mathematical formalism that allows durative actions with MDPs (see, e.g., (Weld et al. 2008)). For the partially observable case, but without constraints, durative actions are also known as macro-actions (Theocharous and Kaelbling 2004), and the model is extended more recently to multi-agent settings (Omidshafiei et al. 2015, 2016). In this work, we consider POMDPs with constraints and stochastic duration functions, which, to the best of our knowledge, have not been studied together.

Planning Horizon. In several planning domains, it is reasonable to assume a finite horizon without discounting rewards. For instance, in domains where utility should be maximized within the next 24 hours, one would be interested in the reward collected during that period, and the utility afterward is no longer relevant. Such situations occur in many applications, e.g., smart power grids, where charging providers solve planning problems with a finite horizon for electric vehicles (Khonji, Chau, and Elbassioni 2018), and unit commitment (Morales-España, Latorre, and Ramos 2013), and demand-response (Khonji, Chau, and Elbassioni 2019) require planning under uncertainty for creating finite-horizon schedules for power supplies and demands. Another example is condition-based maintenance, where, for example, a machine's condition is partially observable at the time of planning (Byon and Ding 2010). As in (Santana, Thiébaux, and Williams 2016), we consider a finite planning horizon which we believe is natural in many risk-sensitive applications.

Our work has two main contributions. First, to the best of our knowledge, it is the first time that a unified model is proposed for POMDP classes with constraints and durative actions. A novel ILP formulation is presented for the unified POMDP framework, which can be solved using existing solvers in the ILP literature for optimal policies. Second, a heuristic forward approach that effectively prunes the search space via solving successive partial ILP programs. The linear relaxation of the ILP, which can be solved faster than ILP, can be used to obtain stochastic policies for applications with no risk constraints. Finally, evaluation results show that our approach is superior to the state-of-the-art fixed-horizon CC-POMDP solver.

2 Problem Definition

A fixed-horizon POMDP M is formally defined as a tuple $M = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, U, b_0, h \rangle$, where $\mathcal{S}, \mathcal{A}$ and $\mathcal{O}$ are finite sets of discrete states, actions and observations; $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a probabilistic transition function between states, $T(s, a, s') = \Pr(s' \mid a, s)$, where $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$; $O : \mathcal{O} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ is a probabilistic observation function such that $O(o, s, a) = \Pr(o \mid s, a)$; $U : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a utility function; $b_0 : \mathcal{S} \to [0, 1]$ is an initial belief state, a probability distribution over $\mathcal{S}$; and h is the planning horizon.

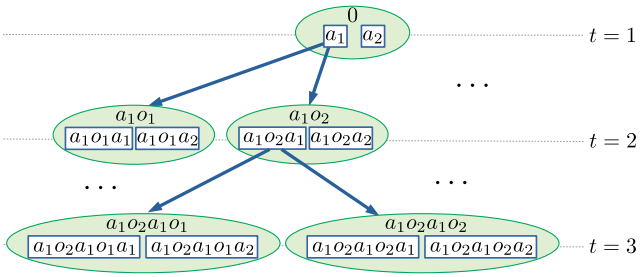
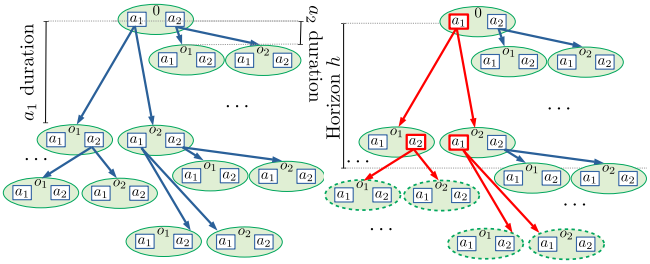


Figure 1: An And-Or tree of all histories. Ellipses represent the set $\widetilde{\mathcal{O}}$, whereas rectangles represent the set $\widetilde{\mathcal{A}}$.



(a) (b)

Figure 2: (a) An And-Or tree of all histories, where edges represent the duration of respective actions. Green-shaded circles are observation sequences $\widetilde{\mathcal{O}}$, while rectangles are action sequences $\widetilde{\mathcal{A}}$, represented by the last action of the sequence. Here, action arrows, as shown in Fig. 1, are removed to emphasize the action duration as arrow length. (b) A policy is represented by the red rectangles. Leaf nodes are represented by dashed circles.

We present some notations that will be used throughout the paper. Denote an action-observation sequence (also called a history) as $q = \langle (a_q^1, o_q^1), (a_q^2, o_q^2), \ldots \rangle$, where $a_q^i \in \mathcal{A}$, $o_q^i \in \mathcal{O}$, and $i \in \mathcal{T}(q) \triangleq \{0, 1, 2, \ldots\}$. The set $\mathcal{T}(q)$ denotes the execution steps of q. A sequence q could end with an observation, i.e., $q = \langle (a_q^1, o_q^1), ..., (a_q^k, o_q^k) \rangle$, or end with an action, $q = \langle (a_q^1, o_q^1), ..., (a_q^k) \rangle$. We write $q = 0$ to indicate the empty sequence. Let $\widetilde{\mathcal{A}}$ be the set of all possible sequences that end with an action, and $\widetilde{\mathcal{O}}$ be the set of all sequences that end with an observation, including the empty sequence. With the two sets, we can obtain an And-Or tree representation of all possible history traces, as shown in Fig. 1. For brevity, we drop the superscript k to indicate the last action (resp. observation) in history q, i.e., $a_q = a_q^k$ (resp. $o_q = o_q^k$). We write $|q| \triangleq |\mathcal{T}(q) \backslash \{0\}|$ to indicate the

length of the history. Also, we say $q \leq q'$ if and only if q precedes q'. In other words, q is a parent node of q' in the And-Or tree. We also assume that q precedes itself, i.e., $q \leq q$. We write $q - k$ to denote history q minus the last k action-observation pairs. If the sequence ends with an action, then $q - 1$ removes the last action. For an observation sequence $q \in \widetilde{\mathcal{O}}$, we write $\overline{q} \in \widetilde{\mathcal{A}}$ to denote the preceding action sequence in the same step. Also, we write $\overline{q} - k$ to indicate the corresponding action node.

A deterministic history-dependent policy, denoted as $\pi(\cdot)$, is a function that maps a history ending with an observation to an action, i.e., $\pi : \widetilde{\mathcal{O}} \to \mathcal{A}$. In the context, $\pi(\cdot)$ represents a policy tree, where the function assigns an action for each element within a subset of $\widetilde{\mathcal{O}}$ that forms a tree structure. We write $\widetilde{\mathcal{O}}_\pi$ to indicate policy tree nodes (see Fig. 2b for an illustration). The objective of POMDP is to compute a policy (or a conditional plan) π^* that maximizes (resp. minimizes) the cumulative expected utility (resp. cost),

$$\pi^* = \arg\max_{\pi} \mathbb{E}\Big[\sum_{q \in \widetilde{\mathcal{O}}_\pi : |q| < h} U(S_q, \pi(q)) \mid \pi\Big], \tag{1}$$

where S_q is a random state at time $|q|$ obtained by following the action-observation sequence q[1]. Notice that the expectation is computed over the random next states induced by the transition and observation models. The problem can be similarly defined with a cost minimization objective, and all the paper's results can be easily adapted for this case.

(Chance) Constrained POMDP

Constrained POMDP (C-POMDP) is defined as a tuple $M' = M \parallel \langle P, C \rangle$, where M is a POMDP model; $P : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a cost function; and $C \in \mathbb{R}$ is an upper bound on the total expected cost. The objective is to compute a policy π^* that maximizes the cumulative expected utility, given by Eqn. (1), while keeping the cumulative expected cost at most C,

$$\mathbb{E}\Big[\sum_{q \in \widetilde{\mathcal{O}}_\pi : |q| < h} P(S_q, \pi^*(q)) \mid \pi^*\Big] \leq C. \tag{2}$$

Chance-constrained POMDP (CC-POMDP) is defined as a tuple $M'' = M \parallel \langle \mathcal{R}, \Delta \rangle$, where $\mathcal{R} \subset \mathcal{S}$ is a subset that represents risky states[2]; and Δ is the risk budget, a threshold on the probability of failure. The objective is to compute a conditional plan that maximizes the cumulative expected utility, given by Eqn. (1) such that the execution risk is at most Δ. The execution risk at the root history $q = 0$ onwards (or at any other history points $q \neq 0$) should satisfy

$$er(q \mid \pi) \triangleq \Pr\Big(\bigvee_{q' \in \widetilde{\mathcal{O}}_\pi : q' \geq q, |q'| \leq h} S_{q'} \in \mathcal{R} \mid q, \pi\Big) \leq \Delta \tag{3}$$

Intuitively, the execution risk is the probability of traversing through risky states throughout any run (Santana, Thiébaux, and Williams 2016; Santana and Williams 2015).

[1]We use capital letters for random variables and small letters for set elements.

[2]One can slightly generalize the model such that risk is defined as the probability of failure $R(s, a, s') \in [0, 1]$ after taking action a from state s and ending in state s'.

The definitions of C-POMDP and CC-POMDP can be easily extended to allow multiple constraints, but for clarity, we will consider only one constraint.

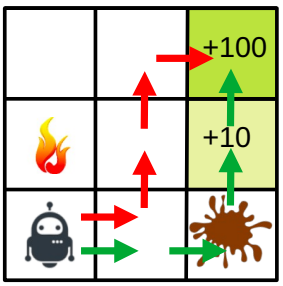


Figure 3: The robot's goal is to collect as much utility as possible. The fire symbol indicates a hazardous state that the robot should avoid, while the mud splash symbol represents a state where navigation may take longer than usual.

Durative Actions

We extend the (C)C-POMDP models with durative actions, defined by a duration function $D(s, a)$. We consider the following duration models.

Fixed duration: Each action a takes an execution time of $c_a \in \mathbb{R}_+$ regardless of the state where the action is taken. The duration function is defined by $D(s, a) = c_a$, for all $s \in \mathcal{S}$. See Fig. 2a for a pictorial illustration. A feasible policy π should satisfy that all leaf nodes in the policy tree incur a total duration of at least h. More precisely, let $\mathcal{L}^\pi \subseteq \widetilde{\mathcal{A}}$ be the set of leaf nodes of policy π, such that for all $q' \in \mathcal{L}^\pi$, $\sum_{q \in \widetilde{\mathcal{O}}, q < q'} c_{\pi(q)} \geq h$. See Fig. 2b for a pictorial illustration. Note that the standard POMDP model can be captured by $D(s, a) = 1, \forall s, a$.

Stochastic duration with percentile risk criteria: The stochastic model considers cases where the duration parameter cannot be modeled accurately. Instead, we use a distribution to capture the underlying uncertainty in the system that is not captured in the model abstraction. The stochastic model has been considered in the literature in the context of MDPs (Delage and Mannor 2010). In this model, we allow noise in the duration function $D(s, a)$, i.e., the duration is a probability distribution with a known distribution rather than a fixed value. For simplicity, we consider certain uni-modal distributions, as illustrated later. For this model, a feasible policy should satisfy

$$\tau(q') \triangleq \Pr\Big(\mathbb{E}\Big[\sum_{q \in \widetilde{\mathcal{O}}_\pi : q < q'} D(S_q, \pi(q)) \mid q'\Big] < h\Big) \leq \varsigma, \tag{4}$$

at all leaf nodes $q' \in \mathcal{L}^\pi \triangleq \{q \in \widetilde{\mathcal{O}} \mid \tau(q) \leq \varsigma\}$, where ς is an input parameter that bounds the tail probability. Intuitively, $\tau(q')$ is the probability of the expected total duration of history q' being less than the planning horizon h, primarily due to parameter uncertainty.

Chance-constrained duration: In this setting, $D(s, a) \in \mathbb{R}_+$ is deterministic. The goal is to have $\tau(q') \triangleq \Pr\Big(\sum_{q \in \widetilde{\mathcal{O}} : q < q'} D(S_q, \pi(q)) < h \mid q'\Big) \leq \varsigma$, for all $q' \in \mathcal{L}^\pi$.

The durative extension of C-POMDP is defined by modifying the conditions in Eqns. (1)-(2), replacing $|q| < h$ by $\tau(q) > \varsigma$. More precisely, the durative C-POMDP is defined as

$$(\textsf{C-POMDP}[M', \varsigma]) \quad \max_{\pi} \mathbb{E}\Big[\sum_{q \in \widetilde{\mathcal{O}}_\pi : \tau(q) > \varsigma} U(S_q, \pi(q)) \mid \pi\Big]$$

$$\text{subject to} \quad \mathbb{E}\Big[\sum_{q \in \widetilde{\mathcal{O}}_\pi : \tau(q) > \varsigma} P(S_q, \pi(q)) \mid \pi\Big] \leq C. \quad (5)$$

Similarly, the durative CC-POMDP can be defined as

$$(\textsf{CC-POMDP}[M'', \varsigma]) \quad \max_{\pi} \mathbb{E}\Big[\sum_{q \in \widetilde{\mathcal{O}}_\pi : \tau(q) > \varsigma} U(S_q, \pi(q)) \mid \pi\Big]$$

$$\text{subject to} \quad \Pr\Big(\bigvee_{q \in \widetilde{\mathcal{O}}_\pi : \tau(q-1) > \varsigma} S_q \in \mathcal{R} \mid \pi\Big) \leq \Delta \quad (6)$$

The fixed duration model can be defined by $\tau(q') \triangleq h - \sum_{q \in \widetilde{\mathcal{O}}, q < q'} c_{\pi(q)}$ and $\varsigma = 0$. In the following, we write (C)C-POMDP to refer to the more general durative version of the problem.

Fig. 3 demonstrates an example of fixed duration action and chance constraint models in which a robot navigates a 3×3 grid from its starting position to maximize its commutative expected utility. Two states in the grid pose distinct types of constraints: one state contains a fire symbol that could destroy the robot, while the other state only slows the robot down, represented by a mud splash. Actions take one second to execute in normal states and three seconds in muddy states. The robot transitions to the desired direction deterministically in all states, except state $(3, 1)$ and $(2, 2)$[3] where it moves to the desired direction with 0.9 probability and to the left with 0.1 probability. Here, we assume a fully observable environment. The robot receives a utility of 100 when it reaches the upper right corner, 10 when it reaches the tile below it, and no utility for any other state.

The figure depicts two policies: the green policy avoids the risky state, while the red policy avoids the muddy state. The green policy execution requires 6 seconds, while the red policy only takes 4 seconds. When the risk threshold is $\Delta = 0.1$, the red policy is infeasible because its execution risk of $0.1 + 0.9 \times 0.1 = 0.19$ exceeds the threshold Δ (as per Cons. (6)). However, if we increase the risk threshold to $\Delta = 0.2$ and set the planning horizon to $h = 5.5$, the red policy is feasible, but the green policy can only collect a utility of $0.9 \times 10 = 9$ as $\tau(\text{green trajectory}) = 5.5 - 6 < \varsigma = 0$. In contrast, the red policy becomes optimal as it collects an expected utility of at $0.9 \times 0.9 \times 100 = 81$ because $\tau(\text{red trajectory}) = 5.5 - 4 > \varsigma$.

We note that our definition of the duration model can capture several useful constraints that are not directly related to action duration. For instance, $\tau(\cdot)$ can model the total fuel or energy consumption of a robot that has to remain below capacity h throughout its execution histories q. Another interesting application of the chance-constrained duration model is *goal-reachability* specification for infinite-horizon POMDPs (Ajdarów, Brlej, and Novotnỳ 2022). Here, a feasible policy should reach absorbing goal states s' with a probability of at least $1 - \varsigma$. As such, we can define $D(s', a) = 1$ for all absorbing goals s' and $D(s, a) = 0$ otherwise, and $h = 1$. Accordingly, a policy that satisfies the chance-constrained duration constraint $\tau(q') \leq \varsigma$ at all leaf nodes q' implies that goal states are reached with a probability of at least $1 - \varsigma$. We could also have multiple criteria encoded in the formulation, such that $\tau(q) = (\tau^1(q), \tau^2(q), ...), \varsigma = (\varsigma^1, \varsigma^2, ...)$.

An important distinction between the durative constraints and POMDP constraints is that the durative constraint applies to every policy realization, which is, in a sense, more strict and leads to *absolute* safe operation under all contingencies as discussed in (Axelrod, Kaelbling, and Lozano-Pérez 2018).

3 Integer Linear Programming Formulation

In this section, we present a unified integer programming formulation (ILP) for POMDP and its constrained extensions. For history $q \in \widetilde{O}$ (resp. $q \in \widetilde{\mathcal{A}}$) and action $a \in \mathcal{A}$ (resp. observation $o \in \mathcal{O}$), we write qa (resp. qo) to denote the concatenation $q \| \langle a \rangle$ (resp. $q \| \langle o \rangle$).

We represent a deterministic policy π as a binary decision vector $\mathbf{x} \in \{0, 1\}^*$, such that for history $q \in \widetilde{\mathcal{A}}$, $x_q = 1$ indicates that the last action in q is selected as part of the policy, and $x_q = 0$ otherwise. A policy tree should satisfy the following definition.

Definition 3.1. *$\mathbf{x} \in \{0, 1\}^*$ is a time-bounded conditional plan if it satisfies,*

$$\sum_{a \in \mathcal{A}} x_a = 1, \sum_{a \in \mathcal{A}} x_{qoa} = x_q, \forall q \in \widetilde{\mathcal{A}}, \forall o \in \mathcal{O} \mid \tau(qo) > \varsigma.$$

The first constraint enforces one action to be selected at the root of the And-Or tree, while the second set of constraints enforces exactly one child action at observation nodes (see Fig. 2b). The two constraints enforce a tree structure that satisfies the durative constraint. Note that a finite number of variables are needed to represent a policy tree according to the above definition. We need a set of variable x_q only for histories $q - 1$ such that $q - 1$ violates the durative constraints, $\tau(q - 1) > \varsigma$. If $q - 1$ satisfies the constraint, then no further actions need to be taken; hence, x_{q-1} should not be part of the plan, as depicted in Fig. 2b for the fixed duration case.

As we will show in the next subsection, the durative (C)C-POMDP can be modeled by the following integer linear program (ILP), with input parameters ς, u_q, r_q, R:

$$(\textsf{ILP}[\varsigma, u_q, r_q, R]) \quad \max_{x_q \in \{0,1\}} \sum_{q \in \widetilde{\mathcal{A}} : \tau(q-1) > \varsigma} u_q \cdot x_q,$$

$$\text{subject to} \sum_{q \in \widetilde{\mathcal{A}} : \tau(q-1) > \varsigma} r_q \cdot x_q \leq R, \quad \sum_{a \in \mathcal{A}} x_a = 1, \quad (7)$$

$$\sum_{a \in \mathcal{A}} x_{qoa} = x_q, \quad \forall q \in \widetilde{\mathcal{A}}, \forall o \in \mathcal{O}, \text{ s.t. } \tau(qo) > \varsigma \quad (8)$$

[3] (x, y) represents the robot position, where x denotes the row number (counting from the top), and y denotes the column number (counting from the left), both starting from one. Here, we assume an episode terminates at a risky state, but such an assumption is not needed in general in CC-POMDP formalism.

Algorithm 1 `Preprocess`$[M', \varsigma]$

Input: C-POMDP model M'; a percentile threshold ς
Output: ILP constants $(u_q, r_q)_{q \in \widetilde{\mathcal{A}}: \tau(q-1) > \varsigma}$, R, and durative function $\tau(\cdot)$

Initialize: $\mathcal{G} \leftarrow \varnothing$; $\mathcal{N} \leftarrow \{0\}$; $\mathcal{F} \leftarrow \varnothing$; $\widetilde{b}_0 \leftarrow \mathbf{0}$; $\bar{b}_0 \leftarrow \mathbf{0}$; $\widetilde{\rho}(q) \leftarrow 0$; $R \leftarrow C$
do
  $q \leftarrow$ Pick an arbitrary element from $\mathcal{N}$; $\mathcal{N} \leftarrow \mathcal{N} \backslash \{q\}$
  for $a \in \mathcal{A}$ **do**
    Obtain $u_{qa}, r_{qa}, \bar{b}_{qa}, \left(\widetilde{b}_{qao}, \tau(qao), \rho(qao)\right)_{o \in \mathcal{O}}$ by
    `Expand`$[qa, \bar{b}_q, (\widetilde{b}^i_q)_{i \in \mathcal{T}(q)}, \rho(q), M']$
    for $o \in \mathcal{O}$ **do**
      if $\tau(qao) > \varsigma$ **then**
        $\mathcal{N} \leftarrow \mathcal{N} \cup \{qao\}$; $\mathcal{F} \leftarrow \mathcal{F} \cup \{qa\}$
while $\mathcal{N} \neq \varnothing$
return $(u_q, r_q)_{q \in \mathcal{F}}, R, \tau(\cdot)$

The ILP is a generalization of the Dual LP of MDP (d'Epenoux 1963), often interpreted as *flow problem*. We show in Sec. 3 below how to derive the duration function $\tau(\cdot)$, and constants u_q, r_q, R such that the ILP is equivalent to (C)C-POMDPs with durative actions. The steps are summarized for C-POMDP in Alg. 1, denoted by `PreProcess`. The pseudocode can be easily modified to account for CC-POMDP as well (by computing *safe* beliefs as shown in Sec. 3).

Input Preprocessing

In this section, we show how to obtain the constants in ILP, namely, utility u_q, penalty or risk r_q, and duration $\tau(q)$, for every history $q \in \widetilde{\mathcal{A}}$ such that $\tau(q-1) > \varsigma$.

Utility and Penalty for C-POMDP The utility u_q and penalty p_q of the last action in a sequence $q \in \widetilde{\mathcal{A}}$ can be calculated following a similar approach to (Aras et al. 2007) with a slight modification to match our settings. Here we define a utility (resp. penalty) value to the last action instead of the whole sequence. Essentially, the utility (resp. penalty) is the product of the probability of sequence q occurring, denoted by $\rho(q)$, and the expected utility (resp. penalty) of the last action a_q in that sequence. We can recursively compute $\rho(q)$ as follows,

$$\rho(q) \triangleq \Pr\left(\bigwedge_{i \in \mathcal{T}(q)} o^i_q \;\middle|\; b_0, \bigwedge_{j \in \mathcal{T}(q)} a^j_q\right) = \prod_{i \in \mathcal{T}(q)} \Pr\left(o^i_q \;\middle|\; b_0, \bigwedge_{\substack{j \in \mathcal{T}(q) \\ j < i}} (a^j_q, o^j_q), a^i_q\right)$$
$$= \prod_{i \in \mathcal{T}(q)} \Pr(o^i_q \mid \bar{b}^i_q), \tag{9}$$

where $\bar{b}^i_q$ is the prior belief state after action a^i_q in q, defined by $\bar{b}^i_q(s) \triangleq \sum_{s' \in \mathcal{S}} T(s', a^i_q, s) \cdot \widetilde{b}^{i-1}_q(s')$. The posterior belief is given by

$$\widetilde{b}^i_q(s) \triangleq \frac{O(o^i_q, s, a^i_q) \cdot \bar{b}^i_q(s)}{\Pr(o^i_q \mid \bar{b}^i_q)}, \quad \forall s \in \mathcal{S}, \tag{10}$$

where $\Pr(o^i_q \mid \bar{b}^i_q) \triangleq \sum_{s \in \mathcal{S}} \bar{b}^i_q(s) \cdot O(o^i_q, s, a^i_q)$. For $|q| = 1$, which is a sequence with a single action, the probability $\rho(q) = 1$.

Thus, the durative C-POMDP can be defined as ILP according to the following lemma, which a generalization (Khonji, Jasour, and Williams 2019) to account for a duration function.

Lemma 3.2 ((Aras et al. 2007))**.** *Durative C-POMDP is equivalent to* ILP *with the following parameters:* $u_q \triangleq \rho(q) \cdot \sum_{s \in \mathcal{S}} \widetilde{b}_{q-1}(s) U(s, a_q)$, $r_q \triangleq \rho(q) \cdot \sum_{s \in \mathcal{S}} \widetilde{b}_{q-1}(s) P(s, a_q)$, $R \triangleq C$.

Remark 1. *An optimal stochastic policy for C-POMDP can be obtained as follows: Define linear relaxation of* ILP*, replacing* $x_q \in \{0, 1\}$ *by* $x_q \in [0, 1]$. *Let* $\bar{\mathbf{x}}$ *be the optimal solution of the linear relaxation. A stochastic policy* $\bar{\pi}$ *is de-*

Algorithm 2 `Expand`$[qa, \bar{b}_q, (\widetilde{b}^i_q)_{i \in \mathcal{T}(q)}, \widetilde{\rho}(q), M']$

Input: History q followed by action $a \in \mathcal{A}$; prior belief $\bar{b}_q$; posterior beliefs $\widetilde{b}^i_q$; probability of occurrence $\widetilde{\rho}$(q); C-POMDP model M'
Output: Utility u_{qa}, risk r_{qa}, prior belief $\bar{b}_{qa}$; duration function $\tau(qao)$, probability $\rho(qao)$, belief $\widetilde{b}_{qao}, \forall o \in \mathcal{O}$

Initialize: $\eta \leftarrow 0$; $\widetilde{b}_{qao} \leftarrow \mathbf{0}$; $\widetilde{\widetilde{b}}^i_q \leftarrow \mathbf{0}$, $f^i_q \leftarrow \mathbf{0}$ for $i \in \mathcal{T}(q)$; $\alpha \leftarrow 0$;
$\bar{b}_{qa} \leftarrow \sum_{s' \in \mathcal{S} \backslash \mathcal{R}} T(s', a, s) \widetilde{b}_q(s')$
$u_{qa} \leftarrow \rho(q) \cdot \sum_{s \in \mathcal{S}} \widetilde{b}_q(s) U(s, a)$
$r_{qa} \leftarrow \rho(q) \cdot \sum_{s \in \mathcal{S}} \widetilde{b}_q(s) P(s, a)$
for $o \in \mathcal{O}$ **do**
  for $s \in \mathcal{S}$ **do**
    $\widetilde{b}_{qao}(s) \leftarrow O(o, s, a) \bar{b}_{qa}(s)$; $\eta \leftarrow \eta + \widetilde{b}_{qao}(s)$
  $\widetilde{b}_{qao}(s) \leftarrow \frac{\widetilde{b}_{qao}(s)}{\eta}, \forall s \in \mathcal{S}$
  $\rho(qao) \leftarrow \eta \cdot \rho(q)$
  for $i = |qao|$ **downto** 0 **do** ▷ Duration function comp.
    Let $q' = qao$
    if $i = |q'|$ **then**
      $f^i_{q'}(s) \leftarrow 1$, for all $s \in \mathcal{S}$
    else
      $f^i_{q'}(s) \leftarrow \sum_{s' \in \mathcal{S}} f^{i+1}_{q'}(s') O(o^{i+1}_{q'}, s', a^i_{q'}) T(s, a^i_{q'}, s')$
    for $s \in \mathcal{S}$ **do**
      $\widetilde{\widetilde{b}}^i_{q'}(s) \leftarrow \widetilde{b}^i_{q'}(s) \cdot f^i_{q'}(s)$; $\alpha \leftarrow \alpha + \widetilde{\widetilde{b}}^i_{q'}(s)$
    $\widetilde{\widetilde{b}}^i_{q'}(s) \leftarrow \frac{\widetilde{\widetilde{b}}^i_{q'}(s)}{\alpha}, \forall s \in \mathcal{S}$
  Fixed Duration: If we assume this model, then each action $a^i_{q'}$ has a fixed duration c_{a^i}. Therefore,
$$\tau(q') \leftarrow \sum_{i \in \mathcal{T}(q')} \sum_{s \in \mathcal{S}} \widetilde{\widetilde{b}}^i_{q'}(s) c_{a^i}$$
  Stochastic Duration with Percentile Risk Criteria: Under this model, and consider the case where $D(s, a) \sim \mathcal{N}(\mu_{s,a}, \sigma^2_{s,a})$ follows a uni-modal Gaussian distribution. Then,
$$\tau(q') \leftarrow \tfrac{1}{2}\left[1 + \operatorname{erf}\left(\frac{h - \sum_{i \in \mathcal{T}(q')} \sum_{s \in \mathcal{S}} \widetilde{\widetilde{b}}^i_{q'}(s) \mu_{s,a}}{\sqrt{2} \sum_{i \in \mathcal{T}(q')} \sum_{s \in \mathcal{S}} \widetilde{\widetilde{b}}^i_{q'}(s)^2 \sigma^2_{s,a}}\right)\right],$$
  where $a \triangleq a^i_{q'}$.
return $u_{qa}, r_{qa}, \bar{b}_{qa}, \left(\widetilde{b}_{qao}, \tau(qao), \rho(qao)\right)_{o \in \mathcal{O}}$

fined as,

$$\bar{\pi}(qa) \triangleq \begin{cases} \frac{\bar{x}_{qa}}{\bar{x}_{\overline{q}}} & \text{if } x_{q-1} > 0 \\ 0 & \text{if } x_{q-1} = 0, \end{cases} \quad \text{for all } a \in \mathcal{A},$$

where $q \in \widetilde{\mathcal{O}}$.

To demonstrate the feasibility of the above policy, let $X_{q'}$ be a random variable of choosing action node $q' \in \widetilde{\mathcal{A}}$. For $q \in \widetilde{\mathcal{O}}$, $X_{qa} = 1$ only when q's parent action is also selected, $X_{\overline{q}} = 1$. One can see that the definition gives a valid probability distribution at node q as it sums up to one. Note that the stochastic policy $\bar{\pi}(qa) = \Pr(X_{qa} = 1 \mid X_{\overline{q}} = 1)$. By the law of total probability,

$$\Pr(X_{qa} = 1) = \Pr(X_{qa} = 1 \mid X_{\overline{q}} = 1) \cdot \Pr(X_{\overline{q}})$$
$$= \tfrac{\bar{x}_{qa}}{\bar{x}_{\overline{q}}} \cdot \Pr(X_{\overline{q}}) = \tfrac{\bar{x}_{qa}}{\bar{x}_{\overline{q}}} \cdot \tfrac{\bar{x}_{\overline{q}}}{\bar{x}_{\overline{q}-1}} \cdot \tfrac{\bar{x}_{\overline{q}-1}}{\bar{x}_{\overline{q}-2}} \cdots = \bar{x}_{qa}.$$

Therefore, the expected total utility is equivalent to that of the linear relaxation of ILP that solves a stochastic policy.

Utility and Risk for CC-POMDP. For a given belief b, let $r(b) \triangleq \sum_{s \in \mathcal{R}} b(s)$ be the probability of being in a risky state. Following the same lines of (Santana, Thiébaux, and Williams 2016), we recursively compute *safe* beliefs,

$$\bar{b}_q(s) \triangleq \frac{\sum_{s' \in \mathcal{S} \setminus \mathcal{R}} T(s', a_q, s) \widetilde{b}_{q-1}(s')}{1 - r(\widetilde{b}_{q-1})},$$
$$\widetilde{b}_q(s) \triangleq \frac{O(o_q, s, a_q) \cdot \bar{b}_q(s)}{\eta}, \widetilde{\Pr}(o \mid \bar{b}_q) \triangleq \eta, \qquad (11)$$

for $q \in \widetilde{\mathcal{O}}$, where η is a normalization factor. The lemma below is an extension of (Khonji, Jasour, and Williams 2019) to account for the duration function.

Lemma 3.3 ((Khonji, Jasour, and Williams 2019))**.** CC-POMDP *is equivalent to* ILP *with the following parameters:* $R \triangleq \Delta - r(b_0)$, $r_q \triangleq \widetilde{\rho}(q) \cdot r(\bar{b}_q), q \in \widetilde{\mathcal{A}}$, $u_q \triangleq \rho^*(q) \cdot \sum_{s \in \mathcal{S}} \widetilde{b}^*_{q-1}(s) U(s, a_q)$, *for* $q \in \widetilde{\mathcal{A}}$, *where* $\widetilde{\rho}(q) \triangleq \prod_{i \in \mathcal{T}(q)} \left(1 - r(\widetilde{b}^{i-1}_q)\right) \cdot \widetilde{\Pr}(o^i_q \mid \bar{b}^i_q)$, $\rho^*(q)$ *and* $\widetilde{b}^*_{q-1}$ *are given by Eqn. (9) and Eqn. (10), respectively.*

Lemma 3.3 also shows that the execution risk in CC-POMDP is linear in $\mathbf{x}$, which enables us to formulate the problem as ILP.

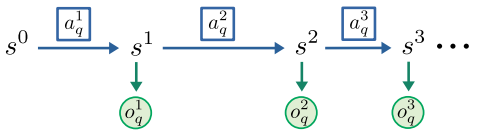


Figure 4: A hidden Markov chain is defined by a sequence of action-observation pairs.

Stochastic Duration Model. We first consider the stochastic duration model, described in Sec. 2, as it captures the fixed duration model as well. We write $s^i \in \mathcal{S}$ to indicate a random state at execution step $i \in \mathcal{T}(q)$. Note that given a fixed history q, the intermediate states s^i resemble a hidden Markov chain, as shown in Fig. 4. The total duration $\tau(q)$, given in Eqn. (4), can be written as,

$$\tau(q) = \Pr\Big(\sum_{i \in \mathcal{T}(q)} \sum_{s^i \in \mathcal{S}} \Pr(s^i \mid q) D(s^i, a^i_q) < h \Big) \qquad (12)$$

$\Pr(s^i \mid q)$ represents a *smoothed* belief at step $i \in \mathcal{T}(q)$, given *future* action-observation pairs a^j_q, o^j_q, $j > i$ throughout q. We denote the smoothed belief by $\widetilde{\bar{b}}^i_q$. Therefore, using Bayes' rule and conditional independence, $\widetilde{\bar{b}}^i_q(s^i) = \Pr(s^i \mid q) = \Pr(s^i \mid q_{\leq i}, q_{>i}) = \frac{1}{\alpha} \widetilde{b}^i_q(s^i) \Pr(q_{>i} \mid s^i)$,where:

$q_{>i} \triangleq \langle (a^j_q, o^j_q) \rangle_{j \in \mathcal{T}(q): j > i}$, $q_{\leq i} \triangleq \langle (a^j_q, o^j_q) \rangle_{j \in \mathcal{T}(q): j \leq i}$,

and α is a normalization factor. When $i = |q|$, $\Pr(q_{>i} \mid s^i) = 1$. For $i < |q|$,

$$\Pr(q_{>i} \mid s^i) = \sum_{s^{i+1}} \Pr(q_{>i} \mid s^{i+1}, s^i) \Pr(s^{i+1} \mid s^i)$$
$$= \sum_{s^{i+1}} \Pr(o^{i+1}, q_{>i+1} \mid s^{i+1}) \Pr(s^{i+1} \mid s^i)$$
$$= \sum_{s^{i+1}} \Pr(q_{>i+1} \mid s^{i+1}) \Pr(o^{i+1}_q \mid s^{i+1}) \Pr(s^{i+1} \mid s^i)$$
$$= \sum_{s^{i+1}} \Pr(q_{>i+1} \mid s^{i+1}) O(o^{i+1}_q, s^{i+1}, a^{i+1}_q) T(s^i, a^i_q, s^{i+1})$$

Thus, the first term of the summation is a recursive call. Therefore, $\Pr(s^i_q \mid q)$ can be computed recursively from the bottom up, as shown in Alg. 2 (Lines 12-18).

Note that Eqn. (12) is a linear transformation of random variables $D(\cdot, \cdot)$ of the same uni-model distribution class. Several distribution classes, such as Normal, Cauchy, and gamma distributions, attain the same distribution after a linear transformation. For instance, suppose $D(s, a) \sim \mathcal{N}(\mu_{s,a}, \sigma^2_{s,a})$ is an independent Gaussian random variable. Let $Q \triangleq \sum_{i \in \mathcal{T}(q)} \sum_{s \in \mathcal{S}} \widetilde{\bar{b}}^i_q(s) D(s, a^i_q)$. Since the noise of each action is assumed to be an independent random variable following a Gaussian distribution, we have $Q \sim G(\overline{\mu}_q, \overline{\sigma}^2_q)$, where $\overline{\mu}_q \triangleq \sum_{i \in \mathcal{T}(q)} \sum_{s \in \mathcal{S}} \widetilde{\bar{b}}^i_q(s) \mu_{s, a^i_q}$, $\overline{\sigma}^2_q \triangleq \sum_{i \in \mathcal{T}(q)} \sum_{s \in \mathcal{S}} \widetilde{\bar{b}}^i_q(s)^2 \sigma^2_{s,a}$ is the variance. Therefore, $\tau(q) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{h - \overline{\mu}_q}{\overline{\sigma}_q \sqrt{2}}\right)\right]$, where erf$(\cdot)$ is the Gauss error function, which can be computed efficiently using numerical techniques.

Chance-constrained Duration Model. The model is based on a generalization of CC-POMDP constraint to account for constraints that span through a sequence of states, not just a single state. First, we augment the state space by a set $\mathcal{G}(q')$ of all possible values of $\sum_{q < q'} D(s_q, \pi(q))$. Thus the new state space is $S' \triangleq S \times \mathcal{G}(q')$. A risky augmented state $s' = \langle s, g \rangle$ with respect to duration should have $g > h$. A transition function T' is defined such that $T'(\langle s, g \rangle, a, \langle s', g' \rangle) = T(s, a, s')$ if $g' = D(s, a) + g$ and zero otherwise. Therefore, the durative constraint amounts to $\tau(q') = \Pr(S_{q'} \in \mathcal{R}') \leq \varsigma$, where $\mathcal{R}'$ is the set of augmented risky states. Following the lines in (Santana, Thiébaux, and Williams 2016), for a fixed sequence of actions and observations, we can obtain a linear expression for $\tau(q')$. However, one issue is that the size of

$\mathcal{G}(q')$ can be exponential in h. This issue can be alleviated by discretizing the set $\mathcal{G}(q')$, such that it satisfies $\Pr\left(\sum_{q\in\widetilde{\mathcal{O}}:q<q'} D(S_q,\pi(q)) < h+\epsilon \mid q'\right) \leq \varsigma$, for some parameter ϵ. It can be shown that using the discretized set, we augment the state space only by a factor of at most $\frac{|q'|^2}{\epsilon}$ (Alyassi and Khonji 2021). As we show in the experiments, our algorithm is not quite sensitive to larger state space, as the main speed hindrances are the sizes of actions and observations and the planning horizon. Due to the limited space, we defer the details to the full version of the paper.

4 Heuristic Forward Search

In this section, we present a heuristic search approach that can scale with larger problems. ILP can be embedded in a heuristics search scheme, which can effectively prune the search space and significantly reduce the number of variables. An *LP-guided* search was first proposed by (Trevizan et al. 2016) for infinite-horizon constrained MDPs, called i-dual. Our algorithm can be viewed as an extension of i-dual to partially observable domains with durative actions. Our algorithm differs in how it models different classes of constraints on the policy as a whole and each policy realization captured by durative constraints.

We define incrementally larger partial ILP problems along with admissible heuristics to guide the search. We use an admissible heuristic for utility, denoted by h^u_q, and another for risk, denoted by h^r_q. In other words, h^u_q gives an upper bound on the optimal utility accrued from history q onward, while h^r_q gives a lower bound on the total execution risk $er(q)$ for the case of CC-POMDPs. For C-POMDPs, a penalty heuristic can be obtained in a way similar to the utility heuristic. These bounds can be generally obtained using domain knowledge. Another way to obtain h^u_q is to solve an unconstrained fully observable version of the problem (Smith and Simmons 2012). A partial ILP problem is defined by a partial And-Or tree, defined by a set of expanded action nodes, $\mathcal{E} \subseteq \widetilde{\mathcal{A}}$, and a set *frontier* nodes, $\mathcal{F} \subseteq \widetilde{\mathcal{A}}$, that lie on the boundary of the search tree. As such, we define a partial integer linear program (p-ILP) as follows.

$$\text{(p-ILP)} \quad \max_{x_q\in\{0,1\}} \sum_{q\in\mathcal{E}} u_q \cdot x_q + \sum_{q\in\mathcal{F}} h^u_q \cdot x_q,$$

$$\text{subject to} \quad \text{Def. } (3.1),$$

$$\sum_{q\in\mathcal{E}} r_q \cdot x_q + \sum_{q\in\mathcal{F}} h^r_q \cdot x_q \leq C. \qquad (13)$$

The new formulation replaces utility values with admissible utility heuristics in the objective and risk heuristics in Cons. (13) for nodes on the search frontier $\mathcal{F}$.

The heuristic search approach for solving ILP is given by Algorithm 3, called HILP. The algorithm gradually expands the frontier nodes $\mathcal{F}$, starting from the root of the And-Or tree (Lines 3-10). Then, it solves an instance of p-ILP, defined by $\mathcal{E}$ and $\mathcal{F}$, using an ILP solver (line 11). Based on the solution, nodes on the frontier with non-zero x_q are moved to the set of expanded nodes (line 13-15). The algorithm expands the search tree including more observations following

Algorithm 3 HILP$[M, \varsigma,$ Solver$]$

Input: (C)C-POMDP model M; a percentile threshold ς; a Solver for p-ILP instances
Output: A solution $\mathbf{x}$ to ILP

1: **Initialize:** $\mathcal{E} \leftarrow \varnothing, \mathcal{N} \leftarrow \{0\}; \mathcal{F} \leftarrow \varnothing; \mathbf{x} \leftarrow \mathbf{0}$
2: **do**
3: **for** $q \in \mathcal{N}$ **do**
4: $\mathcal{N} \leftarrow \mathcal{N} \backslash \{q\}$
5: **for** $a \in \mathcal{A}$ **do**
6: $u_{qa}, r_{qa}, \overline{b}_{qa}, \left(\widetilde{b}_{qao}, \tau(qao), \widetilde{\rho}(qao)\right)_{o\in\mathcal{O}}$
7: $\leftarrow$ Expand$[qa, \overline{b}_q, (\widetilde{b}^i_q)_{i\in\mathcal{T}(q)}, \widetilde{\rho}(q), M]$
8: **if** $\exists o \in \mathcal{O}$ such that $\tau(qao) > \varsigma$ **then**
9: $\mathcal{F} \leftarrow \mathcal{F} \cup \{qa\}$
10: Evaluate heuristic functions h^u_{qa}, h^r_{qa}
11: $\mathbf{x} \leftarrow$ Solver$\left[\text{p-ILP}\left[(u_q, r_q)_{q\in\mathcal{E}}, (h^u_q, h^r_q)_{q\in\mathcal{F}}\right]\right]$
12: **for** $q \in \mathcal{F}$ **do**
13: **if** $x_q > 0$ **then**
14: $\mathcal{E} \leftarrow \mathcal{E} \cup \{q\}; \mathcal{F} \leftarrow \mathcal{F} \backslash \{q\}$
15: **for** $o \in \mathcal{O}$ **do**
16: **if** $\tau(qo) > \varsigma$ **then**
17: $\mathcal{N} \leftarrow \mathcal{N} \cup \{qo\}$
18: **while** $\mathcal{N} \neq \varnothing$
19: **return** $\mathbf{x}$

q. Those *new* nodes, denoted by $\mathcal{N}$, are added only when the duration constraint is not satisfied (Lines 15-17). Otherwise, no observations are needed as $\tau(q) \leq \varsigma$. The algorithm terminates when there are no more new nodes to expand, $\mathcal{N} \neq \varnothing$.

5 Evaluation Results

In our experiments, we focus on risk-sensitive, durative CC-POMDP. We implemented our algorithms in Python 3, and used a standard solver (Gurobi 9.0.1) for ILP and HILP, running on an Intel core i9-9900k, with 32GB of RAM. We also compare our results to state-of-the-art CC-POMDP solver RAO* (Santana, Thiébaux, and Williams 2016)[4]. We divide the experiments into two parts, HILP vs. RAO* with unit durations, and test different durative action models comparing ILP with HILP. Each scenario is averaged over 25 trials. Below are descriptions of the problems modeled as CC-POMDP and solved using our algorithm:

5×5 ***grid game*** consists of five risky states, four actions (the four directions), an initial state, a goal state, and five muddy states[5]. The initial belief is defined to be at the start position with a probability of one. The agent remains in the same state if it hits a wall (on the grid's boundary), moves in the correct direction with a probability of 0.85 (left and right with 0.075), and only observes the number of adjacent walls (0, 1, or 2). The agent correctly observes the number of walls with a probability of 0.85, and equal probabilities for incorrect observations. All actions are assumed to have a duration of 1 second in the fixed duration model. For the expected and stochastic duration models, we assume a duration of 1 sec-

[4] https://github.com/JarvisIsFriday/RAOStar

[5] In matrix notation, the start state is located at (5,1), the goal is at (1,5), risky states at (1,1), (2,4), (2,5), (4,1), (4,2), and muddy states at (1,4), (2,2), (3,3), (4,5), (5,3), respectively.

		ILP												HILP											
		Obj. val.			Time (s)			n			Act. n			Obj			Time (s)			Exp. n			Exp. %		
h	Δ	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S	F	E	S
3	0.1	8.93	8.93	9.59	0.03	0.03	0.33	785	785	7505	628	628	6004	8.93	8.93	9.58	0.04	0.04	0.07	268	268	388	42.7	42.7	6.5
	0.2	8.16	8.16	8.16	0.03	0.03	0.34							8.16	8.16	8.16	0.02	0.02	0.02	160	160	160	25.5	25.5	2.7
	0.3	8.16	8.16	8.16	0.02	0.03	0.34							8.16	8.16	8.16	0.02	0.02	0.02	172	172	172	27.4	27.4	2.9
4	0.1	9.68	9.66	10.26	0.37	0.38	4.17	9425	8945	75965	7540	7156	60772	9.68	9.66	10.26	0.26	0.15	1.07	1084	664	2380	14.4	9.3	3.9
	0.2	8.24	8.16	8.16	0.36	0.38	4.14							8.24	8.16	8.16	0.09	0.03	0.05	592	208	280	7.9	2.9	0.5
	0.3	8.24	8.16	8.16	0.36	0.38	4.02							8.24	8.16	8.16	0.20	0.03	0.05	892	244	292	11.8	3.4	0.5
5	0.1	10.36	10.33	10.50	5.52	5.19	58.23	113105	97460	738590	90484	77968	590872	10.36	10.33	10.50	2.13	1.52	2.41	4360	2932	4732	4.8	3.8	0.8
	0.2	8.33	8.24	8.24	5.46	5.15	54.66							8.33	8.24	8.24	0.87	0.09	0.09	2980	592	592	3.3	0.8	0.1
	0.3	8.33	8.24	8.24	5.29	5.00	51.79							8.33	8.24	8.24	1.09	0.20	0.20	3868	892	892	4.3	1.1	0.2
6	0.1	-	-	-	-	-	-	1357265	-	-	1085812	-	-	10.86	10.69	10.96	10.55	3.86	12.40	12976	6052	12988	1.2	-	-
	0.2	-	-	-	-	-	-							8.52	8.25	8.26	9.44	0.43	0.47	17608	1768	2068	1.6	-	-
	0.3	-	-	-	-	-	-							8.52	8.25	8.26	24.59	0.52	0.59	31036	2068	2380	2.9	-	-

Table 1: Simulation results of durative actions for 5×5 grid game using different action models: 'F' for the fixed unit duration, 'E' for the expected duration, and 'S' for the stochastic duration.

ond for actions moving the agent in any direction to or from a state without mud, and 2 seconds if the states are muddy. Each action has a unit cost, and the goal is to minimize the number of steps to the goal state.

100×100 ***grid game*** is a larger instance of the 5×5 grid game. It is a 100×100 grid with the same set of actions and observations with the same transition and observation probabilities. The risky and muddy states are scaled up following the same pattern in the 5×5 grid game instance. The start state and the goal state are in the lower-left (100,1) and upper-right (1,100) corners, respectively. The purpose of solving this larger instance is to experimentally show the scalability of our algorithms with respect to the state space.

HILP vs. RAO*

We present the results for solving the 5×5 and 100×100 grid games by HILP, and RAO* algorithms in Table 2. The objective of these games is to minimize the total cost to reach the goal while limiting the probability of falling into risky states to Δ. Table 2 shows that HILP outperforms RAO* with the Manhattan heuristics in terms of average running time. The results are also reflected in the objective value, where HILP achieves better objective values and the number of iterations for every value of Δ for every planning horizon h.

We can notice that HILP scales up relatively well with the increase in the state space with regards to the average running time (25 states vs. 10,000 states). For HILP algorithm, unexpectedly, we can see that the running time is relatively high for the 5×5 grid and that the larger the size of the grid, the lower its running time. This behavior could be because of the configuration of the risky states within the grids. Although the configuration follows the same pattern repeatedly across the grids, the possible paths to the goal state vary between the different grids.

Durative Actions

We consider three duration models in our experiments: fixed-duration (F), state-dependent duration, stochastic *without* percentile risk criteria (E), and stochastic duration *with* percentile risk criteria. Model E amount to the *expected* duration. The expected and stochastic models allowed us to model the grid game more naturally with regard to the action durations. For the fixed duration, we assumed a standard POMDP duration model, which is one unit assigned to all actions. As highlighted in the grid game's description, we assume a muddy state is a state that will slow the agent down when passing through it. For the stochastic duration, we set the threshold ς to 0.3 and model the uncertainty in a single action duration by a Gaussian distribution with a mean of zero and a variance of 0.1 (fixed for all actions).

			HILP				RAO*			
Prob.	h	Δ	Obj.	Time	n	Iter.	Obj.	Time	n	Iter.
5x5	3	0.1	8.93	0.04	268	12	9.75	0.13	337	40
		0.2	8.16	0.02	160	9	8.24	0.05	142	13
		0.3	8.16	0.02	172	9	8.24	0.05	175	15
	4	0.1	9.68	0.26	1084	22	10.78	0.71	1309	135
		0.2	8.24	0.09	592	13	8.33	0.29	550	48
		0.3	8.24	0.20	892	20	8.33	0.33	712	60
	5	0.1	10.36	2.13	4360	41	11.20	4.70	5170	470
		0.2	8.33	0.87	2980	26	8.52	2.94	3073	259
		0.3	8.33	1.09	3868	29	8.52	3.76	4228	353
	6	0.1	10.86	10.55	12976	60	11.66	48.64	25393	2196
		0.2	8.52	9.44	17608	49	9.46	78.40	48844	4100
		0.3	8.52	24.59	31036	76	9.46	85.33	50839	4256
100x100	3	0.1	198.93	0.07	268	12	199.75	0.33	337	40
		0.2	198.16	0.05	160	9	198.24	0.13	142	13
		0.3	198.16	0.05	172	9	198.24	0.14	175	15
	4	0.1	199.68	0.30	1084	22	200.78	1.73	1309	135
		0.2	198.24	0.13	592	13	198.33	0.64	550	48
		0.3	198.24	0.23	892	20	198.33	0.78	712	60
	5	0.1	200.36	2.03	4360	41	201.18	11.30	4978	455
		0.2	198.33	0.69	2776	24	198.42	5.73	2104	178
		0.3	198.33	1.05	3856	29	198.42	6.70	2704	226
	6	0.1	200.85	7.14	13108	54	201.63	52.25	15550	1334
		0.2	198.42	3.87	10996	37	199.35	92.97	28735	2419
		0.3	198.42	8.15	18052	53	198.81	29.24	7477	638

Table 2: Simulation results with heuristics.

We can see in Table 1, for ILP and HILP, the average total time results of the fixed duration model compared to the expected duration model is slightly higher. That is because we expect a reduction in the total number of nodes when using the expected model since the algorithm satisfies the duration constraint faster (because we assigned longer durations for some actions based on states). The same argument applies to the results of the stochastic model. The only difference is that since we assigned a relatively small value to ς (given that the fixed model is equivalent to the stochastic model if all the actions have the same duration in all the states and when $\varsigma = 0.5$), Eqn. (4) will not be satisfied early, thus will allow for more nodes to be expanded. We note that for the HILP, we were able to reach higher horizons, $h = 6, 7, 8$ for the fixed, expected, and stochastic duration models, respectively (and solve the smaller horizons in less running time). However, solving for $h = 8$ is impractical since it takes more than 100 seconds.

6 Conclusion

In this work, we proposed a unified framework for computing POMDP with durative actions and constraints. We presented a novel ILP formulation that solves the problem. In order to improve the running time in practice, we showed that our formulation could be embedded in a heuristic search scheme that prunes the search space without affecting optimality. Simulation results show that our approach outperforms the known fastest CC-POMDP solvers.

References

Ajdarów, M.; Brlej, Š.; and Novotnỳ, P. 2022. Shielding in Resource-Constrained Goal POMDPs. *arXiv preprint arXiv:2211.15349*.

Altman, E. 1999. *Constrained Markov decision processes*, volume 7. CRC Press.

Alyassi, R.; and Khonji, M. 2021. Dual formulation for chance constrained stochastic shortest path with application to autonomous vehicle behavior planning. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 4486–4492. IEEE.

Aras, R.; Dutech, A.; Charpillet, F.; et al. 2007. Mixed Integer Linear Programming for Exact Finite-Horizon Planning in Decentralized Pomdps. In *ICAPS*, 18–25.

Axelrod, B.; Kaelbling, L. P.; and Lozano-Pérez, T. 2018. Provably safe robot navigation with obstacle uncertainty. *The International Journal of Robotics Research*, 37(13-14): 1760–1774.

Byon, E.; and Ding, Y. 2010. Season-dependent condition-based maintenance for a wind turbine using a partially observed Markov decision process. *IEEE Transactions on Power Systems*, 25(4): 1823–1834.

Cassandra, A. R. 1998. A survey of POMDP applications. In *Working notes of AAAI 1998 fall symposium on planning with partially observable Markov decision processes*, volume 1724.

Delage, E.; and Mannor, S. 2010. Percentile optimization for Markov decision processes with parameter uncertainty. *Operations research*, 58(1): 203–213.

d'Epenoux, F. 1963. A probabilistic production and inventory problem. *Management Science*, 10(1): 98–108.

Dolgov, D.; and Durfee, E. 2005. Stationary deterministic policies for constrained MDPs with multiple rewards, costs, and discount factors. In *International Joint Conference on Artificial Intelligence*, volume 19, 1326.

Feinberg, E. A.; and Shwartz, A. 1996. Constrained discounted dynamic programming. *Mathematics of Operations Research*, 21(4): 922–945.

Hoey, J.; Poupart, P.; Boutilier, C.; and Mihailidis, A. 2014. POMDP models for assistive technology. In *Assistive Technologies: Concepts, Methodologies, Tools, and Applications*, 120–140. IGI Global.

Hong, S.; Lee, S. U.; Huang, X.; Khonji, M.; Alyassi, R.; and Williams, B. C. 2021. An anytime algorithm for chance constrained stochastic shortest path problems and its application to aircraft routing. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 475–481. IEEE.

Isom, J. D.; Meyn, S. P.; and Braatz, R. D. 2008. Piecewise Linear Dynamic Programming for Constrained POMDPs. In *AAAI*, volume 1, 291–296.

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.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4: 237–285.

Khonji, M.; Chau, S. C.-K.; and Elbassioni, K. 2018. Combinatorial optimization of electric vehicle charging in ac power distribution networks. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 1–6. IEEE.

Khonji, M.; Chau, S. C.-K.; and Elbassioni, K. 2019. Combinatorial optimization of AC optimal power flow with discrete demands in radial networks. *IEEE Transactions on Control of Network Systems*.

Khonji, M.; Jasour, A.; and Williams, B. 2019. Approximability of Constant-horizon Constrained POMDP. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 5583–5590. International Joint Conferences on Artificial Intelligence Organization.

Kim, D.; Lee, J.; Kim, K.-E.; and Poupart, P. 2011. Point-based value iteration for constrained POMDPs. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Mausam, M.; Benazera, E.; Brafman, R.; Meuleau, N.; and Hansen, E. A. 2005. Planning with Continuous Resources in Stochastic Domains. IJCAI'05, 1244–1251.

Morales-España, G.; Latorre, J. M.; and Ramos, A. 2013. Tight and compact MILP formulation for the thermal unit commitment problem. *IEEE Transactions on Power Systems*, 28(4): 4897–4908.

Omidshafiei, S.; Agha-Mohammadi, A.-A.; Amato, C.; and How, J. P. 2015. Decentralized control of partially observable markov decision processes using belief space macro-actions. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 5962–5969. IEEE.

Omidshafiei, S.; Agha-Mohammadi, A.-A.; Amato, C.; Liu, S.-Y.; How, J. P.; and Vian, J. 2016. Graph-based cross entropy method for solving multi-robot decentralized POMDPs. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 5395–5402. IEEE.

Ono, M.; Kuwata, Y.; and Balaram, J. 2012. Joint chance-constrained dynamic programming. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 1915–1922. IEEE.

Poupart, P.; Malhotra, A.; Pei, P.; Kim, K.-E.; Goh, B.; and Bowling, M. 2015. Approximate linear programming for constrained partially observable markov decision processes. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Santana, P.; Thiébaux, S.; and Williams, B. 2016. RAO*: an algorithm for chance constrained POMDPs. In *Proc. AAAI Conference on Artificial Intelligence*.

Santana, P. H.; and Williams, B. C. 2015. Dynamic execution of temporal plans with sensing actions and bounded risk. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Smith, T.; and Simmons, R. 2012. Heuristic search value iteration for POMDPs. *arXiv preprint arXiv:1207.4166*.

Sondik, E. J. 1971. The Optimal Control of Partially Observable Markov Decision Processes. *PhD the sis, Stanford University*.

Spaan, M. T.; and Spaan, N. 2004. A point-based POMDP algorithm for robot planning. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 3, 2399–2404. IEEE.

Theocharous, G.; and Kaelbling, L. P. 2004. Approximate planning in POMDPs with macro-actions. In *Advances in Neural Information Processing Systems*, 775–782.

Trevizan, F.; Thiébaux, S.; Santana, P.; and Williams, B. 2016. Heuristic search in dual space for constrained stochastic shortest path problems. In *Twenty-Sixth International Conference on Automated Planning and Scheduling*.

Undurti, A.; and How, J. P. 2010. An online algorithm for constrained POMDPs. In *2010 IEEE International Conference on Robotics and Automation*, 3966–3973. IEEE.

Weld, D. S.; et al. 2008. Planning with durative actions in stochastic domains. *Journal of Artificial Intelligence Research*, 31: 33–82.