Probabilistic Planning with Partially Ordered Preferences over Temporal Goals

Hazhar Rahmani, Abhishek N. Kulkarni, and Jie Fu

Abstract— **In this paper, we study planning in stochastic systems, modeled as Markov decision processes (MDPs), with preferences over temporally extended goals. Prior work on temporal planning with preferences assumes that the user preferences form a total order, meaning that every pair of outcomes are comparable with each other. In this work, we consider the case where the preferences over possible outcomes are a partial order rather than a total order. We first introduce a variant of deterministic finite automaton, referred to as a preference DFA, for specifying the user's preferences over temporally extended goals. Based on the order theory, we translate the preference DFA to a preference relation over policies for probabilistic planning in a labeled MDP. In this treatment, a most preferred policy induces a *weak-stochastic nondominated* probability distribution over the finite paths in the MDP. The proposed planning algorithm hinges on the construction of a multi-objective MDP. We prove that a weak-stochastic nondominated policy given the preference specification is Pareto-optimal in the constructed multi-objective MDP, and vice versa. Throughout the paper, we employ a running example to demonstrate the proposed preference specification and solution approaches. We show the efficacy of our algorithm using the example with detailed analysis, and then discuss possible future directions.**

I. Introduction

With the rise of artificial intelligence, robotics and autonomous systems are being designed to make complex decisions by reasoning about multiple goals at the same time. Preference-based planning (PBP) allows the systems to decide which goals to satisfy when not all of them can be achieved [6]. Even though PBP has been studied since the early 1950's, most works on preference-based temporal planning (c.f. [2]) assume that all outcomes are pairwise comparable—that is, the preference relation is a *total* order. This assumption is strong and, in many cases, unrealistic [1]. In robotic applications, preferences may need to admit a *partial* order because of (a) *Inescapability*: An agent has to make decisions under time limits but with partial information about preferences because, for example, it lost communication with the server; and (b) *Incommensurability*: Some situations, for instance, comparing the quality of an apple to that of banana, are fundamentally incomparable since they lack a standard basis to compare. These situations motivate the need for a planner that deals with partial order preferences in the presence of all uncertainties in its environment.

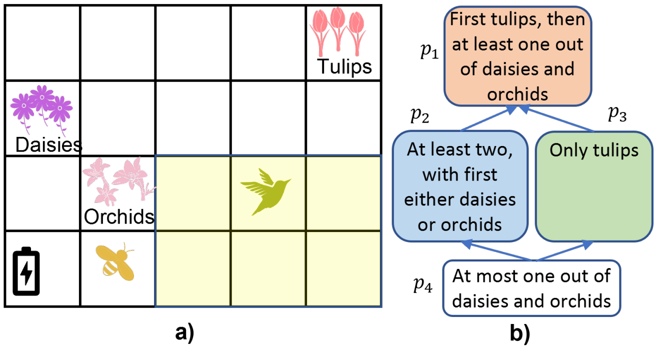
Tulips
Daisies
Orchids
a)
p_1
First tulips, then at least one out of daisies and orchids
p_2
At least two, with first either daisies or orchids
p_3
Only tulips
p_4
At most one out of daisies and orchids
b)

Fig. 1: **a)** Bob's Garden. **b)** Bob's preferences on how the bee robot perform the task of pollinating the flowers.

As a motivation example, consider Figure 1, which shows a garden that belongs to Bob. He grows three kinds of flowers: Tulips, daisies, and orchids. To pollinate the flowers, he uses a bee robot with limited battery. The environment is uncertain due to the presence of another agent (bird), the weather, and the robot dynamics.

Bob has a preference for how the robot should achieve the task of pollination. Compared to the other types, tulips have a shorter life span, so Bob considers four outcomes

(p_1) pollinate tulips first, then at least one other flower type;
(p_2) pollinate two types of flowers, with the first being either daisies or orchids;
(p_3) pollinate only tulips; and
(p_4) at most one out of daisies and orchids is pollinated,

where the preference relation among them is shown in Figure 1b using a preference graph, where the nodes represent the outcomes, and each directed edge is an improving flip [14]. Thus, p_1 is the most preferred and p_4 is the least preferred outcome, while p_2 and p_3 are incomparable with each other. As the robot has a limited battery life and the system is stochastic, it might not achieve the most preferred outcome with probability one. Incomparable outcomes also introduce incomparable policies.

Preference-based planning problems over temporal goals have been well-studied for deterministic planning given both total and partial preferences (see [2] for a survey). For preference specified over temporal goals, several works [12], [15], [16] proposed minimum violation planning methods that decide which low-priority constraints should be violated in a deterministic system. Mehdipour *et al.* [11] associate weights with Boolean and temporal operators in signal temporal logic to specify the importance of satisfying the

[1] The authors are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32605, USA. {h.rahmani, a.kulkarni2, fujie}@ufl.edu This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-21-1-0085 and in part by NSF under award number 2024802.

sub-formula and priority in the timing of satisfaction. This reduces the PBP problem to that of maximizing the weighted satisfaction in deterministic dynamical systems. However, the solutions to PBP problem for deterministic systems cannot be applied to stochastic systems. This is because in stochastic systems, even a deterministic policy yields a distribution over outcomes. Hence, to determine a better policy, we need comparison of distributions—a task a deterministic planner cannot do.

Several works have studied the PBP problem for stochastic systems. Lahijanian and Kwiatkowska [8] considered the problem of revising a given specification to improve the probability of satisfaction of the specification. They formulated the problem as a multi-objective Markov Decision Process (MDP) problem that trades off minimizing the cost of revision and maximizing the probability of satisfying the revised formula. Li *et al.* [9] solve a preference-based probabilistic planning problem by reducing it to a multi-objective model checking problem. However, all these works assume the preference relation to be *total*. To the best of our knowledge, [5] is the only work that studies the problem of probabilistic planning with incomplete preferences. The authors introduce the notion of the value of preference satisfaction for planning within a pre-defined finite time duration and developed a mixed-integer linear program to maximize the satisfaction value for a subset of preference relations. In comparison, our work resorts to the notion of stochastic ordering to compare policies in the stochastic system with respect to the partial order of temporal goals and allows the time horizon to be finite, but unbounded.

To conclude, our contributions in this paper are three-fold: (1) We introduce a new computational model called a *Preference Deterministic Finite Automaton (PDFA)*. A PDFA can model a user's (possibly partial) preferences over temporally extended goals; (2) We identify the connection between the PBP problem in probabilistic planning and stochastic orders [10]. This allows us to reduce the problem of probabilistic planning with partial preferences over temporal goals to that of finding the set of weak-stochastic nondominated policies in a product of MDP and the PDFA. (3) We employ the property of weak-stochastic nondominated policies to design multiple objectives in the product MDP and prove that a Pareto-optimal policy in the resulting multi-objective product MDP is weak-stochastic nondominated respecting the preference relation. Thus, the set of weak-stochastic nondominated policies can, then, be computed using any off-the-shelf solver that computes Pareto optimal policies in polynomial time.

II. Preliminaries and Problem Formulation

Notations The set of all finite words over a finite alphabet Σ is denoted Σ^*. The empty string Σ^0 is denoted as ϵ. The set of all probability distributions over a finite set X is denoted $\mathcal{D}(X)$. Given a distribution $\mathbf{d} \in \mathcal{D}(X)$, the probability of an outcome $x \in X$ is denoted $\mathbf{d}(x)$.

A. The System and its Policy

We model the system using a variant of MDP.

Definition 1. A terminating labeled Markov decision process (TLMDP), or a terminating MDP for short, is a tuple $M = \langle S, A := \bigcup_{s \in S} A_s, \mathbf{P}, s_0, s_\perp, \mathcal{AP}, L\rangle$ in which S is a finite set of states; A is a finite set of actions, where for each state $s \in S$, A_s is the set of available actions at s; $\mathbf{P} : S \times A \to \mathcal{D}(S)$ is the probabilistic transition function, where for each $s, s' \in S$ and $a \in A$, $\mathbf{P}(s, a, s')$ is the probability that the MDP transitions to s' after taking action a at s; $s_0 \in S$ is the initial state; $s_\perp \in S$ is the *termination state*, which is a unique *sink* state and $A_{s_\perp} = \emptyset$; $\mathcal{AP}$ is a finite set of atomic propositions; and $L : S \to 2^{\mathcal{AP}} \cup \{\epsilon\}$ is a labeling function that assigns to each state $s \in S \setminus \{s_\perp\}$, the set of atomic propositions $L(s) \subseteq \mathcal{AP}$ that hold in s. Only the terminating state is labeled the empty string, i.e., $L(s) = \epsilon$ iff $s = s_\perp$.

Though this definition assumes a single sink state, we do not lose generality, as one can always convert any MDP with more than one sink state into an equivalent MDP that has only a single sink state by redirecting proper transitions to that sink state.

The robot's interaction with the environment in a finite number k of steps produces an *execution* $\varrho = s_0 a_0 s_1 a_1 \cdots s_{k-1} a_{k-1} s_k$, where s_0 is the initial state and at each step $0 \leq i \leq k$, the system is at state s_i, the robot performs $a_i \in A_{s_i}$, and then the system transitions to state s_{i+1}, picked randomly based on $\mathbf{P}$ among those states for which $\mathbf{P}(s_i, a_i, .) > 0$. This execution produces a *path* defined as $\rho = s_0 s_1 \cdots s_k \in S^*$, and the *trace* of this path is defined as the finite word $\mathrm{trace}(\rho) = L(s_0)L(s_1)L(s_2)\cdots L(s_k) \in (2^{\mathcal{AP}})^*$. Path ρ is called *terminating* if $s_k = s_\perp$. The set of all terminating paths in M is denoted $\mathrm{Paths}_\perp(M)$.

A policy for M is a function $\pi : \textsc{Dom} \to \textsc{CoDom}$ where it is called *memoryless* if $\textsc{Dom} = S$; *finite-memory* if $\textsc{Dom} = S^*$; *deterministic* if $\textsc{CoDom} = A$, and *randomized* if $\textsc{CoDom} = \mathcal{D}(A)$.

In a terminating MDP, a policy is *proper* if it guarantees that the termination state $s_\perp$ will be reached with probability one [3]. The set of all randomized, finite-memory, proper polices for M is denoted Π^M_{prop}. We are only interested in finite traces for which a preference relation is defined. Thus, we only consider proper policies.

Assumption 1. We assume all the policies for the MDP are proper.

In this paper, we consider only the MDPs for which all the policies are proper. We consider applications where the robot finishes its execution in a finite time, and in fact, in many robotics application, the robot has a battery limit or a limited lifespan and cannot execute forever.

B. Rank the policies

We introduce a computational model that captures the user's preference over different temporal goals.

Definition 2. Given a countable set U, a *preference model* for U, denoted $\succeq^U$ is a partial order over the elements of U.

We simply use $\succeq$ for $\succeq^U$ if its meaning is clear from the context. Given $u_1, u_2 \in U$, we write $u_1 \succeq u_2$ if u_1 is *weakly preferred to* (i.e., is at least as good as) u_2; and $u_1 \sim u_2$ if $u_1 \succeq u_2$ and $u_2 \succeq u_1$, that is, u_1 and u_2 are *indifferent*. We write $u_1 \succ u_2$ to mean that u_1 is *strictly preferred* to u_2, i.e., $u_1 \succeq u_2$ and $u_1 \not\sim u_2$. We write $u_1 \nparallel u_2$ if u_1 and u_2 are *incomparable*.

Definition 3. [10]. Given a countable set U partially ordered by a preference model $\succeq$, the *weak-stochastic ordering* for U is denoted $\mathfrak{E}_{wk}(U)$ and is defined as the family of subsets

$$\mathfrak{E}_{wk}(U) = \bigcup_{x \in U} \{x\}^{\uparrow} \cup \{U, \emptyset\}. \tag{1}$$

where $\{x\}^{\uparrow} = \{y \mid y \succeq x\}$ contains all elements in U that are at least as good as x, according to the partial order $\succeq$.

The weak-stochastic ordering for U allows us to rank different probability measures on U. Given two probability measures P_1 and P_2 on U, we say P_1 *weak-stochastic dominates* P_2 under $\succeq$, denoted $P_1 >_{\mathfrak{E}_{wk}} P_2$, if $P_1[X] \geq P_2[X]$ for each $X \in \mathfrak{E}_{wk}(U)$ and $P_1[Y] > P_2[Y]$ for some $Y \in \mathfrak{E}_{wk}(U)$. Intuitively, for any outcome x in U, the probability of getting an outcome (weakly) preferred to x in P_1 is at least as good as that in P_2, and for some outcome $x' \in U$, the probability of getting an outcome preferred to x' in P_1 is higher than that in P_2.

To illustrate, consider the following example.

Example 1. Let $U = \{a, b, c, d\}$ and $\succeq = \{(a, b), (b, d), (c, d), (a, c), (a, d)\}$, where $(x, y) \in \succeq$ if and only if $x \succeq y$. We have

$$\mathfrak{E}_{wk}(U) = \{\{a\}, \{a, b\}, \{a, c\}, \{a, b, c, d\}, \emptyset\}.$$

Now consider three probability measures P_1, P_2, and P_3 where $P_1(a) = P_1(b) = 0.5$, $P_2(a) = P_2(c) = 0.5$, and $P_3(a) = P_3(d) = 0.5$. Accordingly,

$$[P_1[X]]_{X \in \mathfrak{E}_{wk}(U)} = [0.5, 1, 0.5, 1, 0],$$
$$[P_2[X]]_{X \in \mathfrak{E}_{wk}(U)} = [0.5, 0.5, 1, 1, 0], \text{and}$$
$$[P_3[X]]_{X \in \mathfrak{E}_{wk}(U)} = [0.5, 0.5, 0.5, 1, 0].$$

Therefore, $P_1 >_{\mathfrak{E}_{wk}} P_3$, $P_2 >_{\mathfrak{E}_{wk}} P_3$. None of P_1 and P_2 weak-stochastic dominates the other one.

In this context, the user preference over temporal goals is a preference model for $U = \Sigma^*$ where $\Sigma = 2^{\mathcal{AP}}$. Based on the ranking of probability measures induced by the weak-stochastic ordering for Σ^*, we can rank the proper policies Π^M_{prop} in the TLMDP as follows.

Note that a proper policy $\pi : S^* \to \mathcal{D}(A)$ produces a distribution over the set of all terminating paths in the MDP M such that for each terminating path $\rho \in \mathrm{Paths}_{\perp}(M)$, $\mathbf{Pr}^{\pi}(\rho)$ is the probability of generating ρ when the robot uses policy π. Each terminating path ρ is mapped to a single word in Σ^*, namely $\mathrm{trace}(\rho)$, and therefore, π yields a distribution over the set of all finite words over Σ such that for each word $w \in \Sigma^*$, $\mathbf{Pr}^{\pi}(w)$ is the probability that π produces w.

Definition 4. Given two proper policies π, π' in the terminating labeled MDP M, π *weak-stochastic dominates* π', denoted $\pi >_{\mathfrak{E}_{wk}} \pi'$, if for each $w \in \Sigma^*$, it holds that $\mathbf{Pr}^{\pi}(\{w\}^{\uparrow}) \geq \mathbf{Pr}^{\pi'}(\{w\}^{\uparrow})$, and there exists a word $w' \in \Sigma^*$ such that $\mathbf{Pr}^{\pi}(\{w'\}^{\uparrow}) > \mathbf{Pr}^{\pi'}(\{w'\}^{\uparrow})$.

This definition is used to introduce the following notion.

Definition 5. A proper policy $\pi \in \Pi^M_{prop}$ is *weak-stochastic nondominated* if there *does not exist* any policy $\pi' \in \Pi^M_{prop}$ such that $\pi' >_{\mathfrak{E}_{wk}} \pi$.

Informally, we say a policy π is *preferred*, if and only if it is weak-stochastic nondominated in Π^M_{prop}.

Next, we state our problem informally.

Problem 1. Given a terminating labeled MDP and a preference model $\succeq$ over finite words Σ^*, compute a proper policy that is weak-stochastic nondominated.

III. Main results

A. Preference Deterministic Finite Automaton

In this section, we propose a finite automaton to compatibly represent the user preferences over temporal goals.

Definition 6. A preference deterministic finite automaton (PDFA) for an alphabet Σ is a tuple $\mathcal{A} = \langle Q, \Sigma, \delta, \iota, G := (\mathbb{F}, E)\rangle$ in which Q is a finite set of states; Σ is the alphabet; $\delta : Q \times \Sigma \to Q$ is the transition function; $\iota \in Q$ is the initial state; and $G = (\mathbb{F}, E)$ is a *preference graph* in which, $\mathbb{F} = \{F_1, F_2, \cdots, F_m\}$ is a partition of Q—i.e., $F \subseteq Q$ for each $F \in \mathbb{F}$, $F \cap F' = \emptyset$ for each distinct state subsets $F, F' \in \mathbb{F}$, and $\bigcup_{F \in \mathbb{F}} F = Q$; and $E \subseteq \mathbb{F} \times \mathbb{F}$ is a set of directed edges.

With a slight abuse of notation, we define the extended transition function $\delta : Q \times \Sigma^* \to Q$ in the usual way, i.e., $\delta(q, \sigma w) = \delta(\delta(q, \sigma), w)$ for $w \in \Sigma^*$ and $\sigma \in \Sigma$, and $\delta(q, \epsilon) = q$. Note that Definition 6 augments the classical deterministic finite automaton [7] with the preference graph G, instead of a set of accepting (final) states.

For two vertices $F, F' \in \mathbb{F}$, we write $F \rightsquigarrow F'$ to denote F' is *reachable* from F. By convention, each vertex F of G is reachable from itself. That is, $F \rightsquigarrow F$ always holds.

The PDFA encodes a preference model $\succeq$ for $\Sigma = (2^{\mathcal{AP}})^*$ as follows. Consider two words $w, w' \in \Sigma^*$. Let $F, F' \in \mathbb{F}$ be the two state subsets such that $\delta(q, w) \in F$ and $\delta(q, w') \in F'$ (recall that $\mathbb{F}$ is a partitioning of F); There are four cases: (1) if $F = F'$, then $w \sim w'$; (2) if $F \neq F'$ and $F' \rightsquigarrow F$, then $w \succ w'$; (3) if $F \neq F'$ and $F \rightsquigarrow F'$, then $w' \succ w$; and (4) otherwise, $w \nparallel w'$.

To illustrate, see Figure 2, which shows a preference DFA specifying the preferences in the example of Figure 1. State subsets $F_1 = \{q_2\}$, $F_2 = \{q_4\}$, $F_3 = \{q_1\}$, and $F_4 = \{q_0, q_3, q_5\}$ respectively represent preferences p_1 through p_4.

The following Lemma allows us to define the weak-stochastic ordering over Σ^*, defined by the PDFA, using its preference graph.

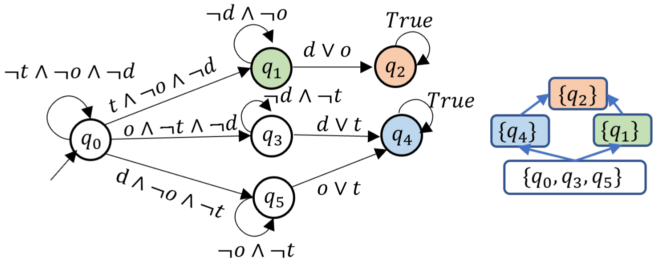

Fig. 2: PDFA for the example in Figure 1. **Left)** The DFA structure of PDFA. **Right)** The preference graph of PDFA.

Lemma 1. For each word $w \in \Sigma^*$, if $\delta(\iota, w) \in F$ for some $F \in \mathbb{F}$, then

$$\{w\}^{\uparrow} = \{w' \in \Sigma^* \mid \exists F' \in \mathbb{F}, \delta(\iota, w') \in F' \text{ and } F \rightsquigarrow F'\} \tag{2}$$

The lemma directly follows from the transitivity property in the transition function in $\mathcal{A}$ and the preference relation and thus the proof is omitted.

Problem: Probabilistic Planning with Partially Ordered Preferences (PPwPOP)

Input: A TLMDP $M = \langle S, A := \Sigma_{s \in S} A_s, \mathbf{P}, s_0, s_\perp, \mathcal{AP}, L\rangle$ and a PDFA $\mathcal{A} = \langle Q, 2^{\mathcal{AP}}, \delta, \iota, G := (\mathbb{F}, E)\rangle$.

Output: The set of all proper polices for M that are weak-stochastic nondominated under the preferences specified by $\mathcal{A}$.

IV. Synthesizing a Preferred Policy

We now present our algorithm. The first step is to augment the planning state space with the state of the PDFA. With this augmented state space, we can relate the preferences over traces in the MDP to a preference over subsets of terminating states in a product MDP we define as follows.

Definition 7 (Product MDP)**.** Let $M = \langle S, A := \Sigma_{s \in S} A_s, \mathbf{P}, s_0, s_\perp, \mathcal{AP}, L\rangle$ and $\mathcal{A} = \langle Q, \Sigma, \delta, \iota, G := (\mathbb{F}, E)\rangle$ be respectively the TLMDP and the PDFA. The product of M and $\mathcal{A}$ is a tuple $\mathcal{M} = (X, A := \bigcup_{x \in X} A_x, \mathbf{T}, x_0, X_G, \mathcal{G} := (\mathcal{W}, \mathcal{E}))$ in which

1) $X = S \times Q$ is the state space;
2) A is the action space, where for each $x = (s, q) \in X$, $A_x = A_s$ is the set of available actions at state x;
3) $\mathbf{T} : X \times A \times X \to [0, 1]$ is the transition function such that for each state $(s, q) \in X$, action $a \in A$, and state $(s', q') \in X$;

$$\mathbf{T}((s,q), a, (s',q')) = \begin{cases} \mathbf{P}(s, a, s') & \text{if } q' = \delta(q, L(s')), \\ 0 & \text{otherwise;} \end{cases} \tag{3.a}$$

4) $x_0 = (s_0, \delta(\iota, L(s_0)))$ is the initial state;
5) $X_G = \{s_\perp\} \times Q$ is the set of terminating states;
6) $\mathcal{G} = (\mathcal{W}, \mathcal{E})$ is the preference graph, in which, letting $W_i = \{s_\perp\} \times F_i$ for each $F_i \in \mathbb{F}$,
   - $\mathcal{W} = \{W_i \mid i = 1, \ldots, |\mathbb{F}|\}$ is the vertex set of the graph, and
   - $\mathcal{E}$ is the edge set of the graph such that $(W_i, W_j) \in \mathcal{E}$ if and only if $(F_i, F_j) \in E$.

The preference graph of this MDP has been directly lifted from the one defined for the PDFA. We use $W \rightsquigarrow W'$ to denote that W' is reachable from W in the preference graph $\mathcal{G}$. Again, every W is reachable from itself.

Continuing with the example in Figure 2, we have $W_1 = \{s_\perp\} \times F_1 = \{(s_\perp, q_2)\}$, $W_2 = \{s_\perp\} \times F_2 = \{(s_\perp, q_4)\}$, $W_3 = \{s_\perp\} \times F_3 = \{(s_\perp, q_1)\}$, and $W_4 = \{s_\perp\} \times F_4 = \{(s_\perp, q_0), (s_\perp, q_3), (s_\perp, q_5)\}$.

Next, we show how to compute a weak-stochastic nondominated policy, in the sense of Definition 5, through solving a multi-objective MDP.

Given the product MDP $\mathcal{M}$ constructed in Definition 7, the *weak-stochastic ordering* for $\mathcal{W}$, denoted $\mathfrak{E}_{wk}(\mathcal{W})$, is the family of subsets

$$\mathfrak{E}_{wk}(\mathcal{W}) = \{\{W\}^{\uparrow} \mid W \in \mathcal{W}\} \cup \{\emptyset, \mathcal{W}\} \tag{3}$$

where $\{W\}^{\uparrow} = \bigcup_{W' \in \mathcal{W}, W \rightsquigarrow W'} \{W'\}$.

Note that by construction, the number of subsets in $\mathfrak{E}_{wk}(\mathcal{W})$ minus the empty set and the set $\mathcal{W}$ is exactly the size of $\mathcal{W}$. Let $N = |\mathcal{W}|$.

Definition 8 (Multi-objective MDP (MOMDP))**.** The multi-objective MDP (MOMDP) associated with the product MDP $\mathcal{M} = \langle X, A, \mathbf{T}, x_0, X_G, \mathcal{G} := (\mathcal{W}, \mathcal{E})\rangle$ in Definition 7 is a tuple $\mathcal{P} = \langle X, A := \bigcup_{x \in X} A_x, \mathbf{T}, x_0, X_G, \mathcal{Z} = \{Z_1, Z_2, \cdots, Z_N\}\rangle$ in which X, A, $\mathbf{T}$, x_0, and X_G are the same elements in $\mathcal{M}$ and for each $i \in \{1, \cdots, N\}$, $Z_i = \bigcup_{W \in \{W_i\}^{\uparrow}} W$. The i-th objective in the MOMDP is to maximize the probability for reaching the set Z_i.

Note that each Z_i is a subset of goal states X_G, and that the intersection of two distinct goal subsets Z_i and Z_j may not be empty.

Using the running example in Figure 2, we have $\{W_1\}^{\uparrow} = \{W_1\}$, $\{W_2\}^{\uparrow} = \{W_1, W_2\}$, $\{W_3\}^{\uparrow} = \{W_1, W_3\}$, and $\{W_4\}^{\uparrow} = \{W_1, W_2, W_3, W_4\}$; $Z_1 = W_1$, $Z_2 = W_1 \cup W_2$, $Z_3 = W_1 \cup W_3$, and $Z_4 = \bigcup_{i=1}^{4} W_i$.

In this MOMDP, for a given randomized, finite-memory policy $\mu : X^* \to \mathcal{D}(A)$, we can compute the value vector of μ as a N-dimensional vector $\mathbf{V}^{\mu} = [\mathbf{V}_1^{\mu}, \mathbf{V}_2^{\mu}, \cdots, \mathbf{V}_N^{\mu}]$ where for each i, $\mathbf{V}_i^{\mu}$ is the probability of reaching states of Z_i by following policy μ, starting from the initial state.

Given a randomized, memoryless policy $\mu : X \to \mathcal{D}(A)$, to compute its value vector $\mathbf{V}^{\mu}$, we first set for each goal state $x_g \in X_G$, $\mathbf{V}^{\mu}(x_g)$ to be the vector such that for each $i \in \{1, \cdots, n\}$, $\mathbf{V}_i^{\mu}(x_g) = 1$ if $x_g \in Z_i$, and otherwise $\mathbf{V}_i^{\mu}(x_g) = 0$. Then we compute the values of the non-goals states $x \in X \setminus X_G$ via the Bellman recurrence

$$\mathbf{V}^{\mu}(x) = \sum_{a \in A} \left(\mu(x)[a] \sum_{x' \in X} \mathbf{T}(x, a, x') \mathbf{V}^{\mu}(x') \right). \tag{4}$$

Definition 9. Given two proper polices μ and μ' for $\mathcal{M}$, it is said that μ *Pareto dominates* μ', denoted $\mu > \mu'$, if for each $i \in \{1, \cdots, N\}$, $\mathbf{V}_i^{\mu} \geq \mathbf{V}_i^{\mu'}$, and for at least one $j \in \{1, \cdots, n\}$, $\mathbf{V}_j^{\mu} > \mathbf{V}_j^{\mu'}$.

Intuitively, μ Pareto dominates μ' if, compared to μ', it increases the probability of reaching at least a set Z_i without reducing the probability of reaching other sets Z_j's.

Definition 10. A proper policy μ for the MOMDP in Definition 8 is *Pareto optimal* if for no proper policy μ' for the MOMDP it holds that $\mu' > \mu$.

In other words, a policy is Pareto optimal if it is not dominated by any policy. The *Pareto front* is the set of all Pareto optimal policies. It is well-known that the set of memoryless policies suffices for achieving the Pareto front [4]. Thus, we restrict to compute memoryless policies.

With this in mind, we present the following result.

Theorem 1. Let $\mu : X \to \mathcal{D}(A)$ be a policy for $\mathcal{P}$. Construct policy $\pi : S^* \to \mathcal{D}(A)$ for the TLMDP M such that for each $\rho = s_0 s_1 \cdots s_n \in S^*$ it is set $\pi(\rho) = \mu(s_n, \text{trace}(\rho))$. If μ is Pareto optimal, then π is weakly-stochastic nondominated, respecting the preference specified by PDFA $\mathcal{A}$.

Proof. We show that if μ is Pareto optimal then π is weak-stochastic nondominated. To facilitate the proof, the following notation is used: Let $\mathbf{Pr}^{\mu}(\text{reach}(X), \mathcal{M})$ be the probability of terminating in the set X given the policy μ for the MOMDP and $\mathbf{Pr}^{\pi}(\text{reach}(X), M)$ be the probability of terminating in the set X given the policy π in the original TLMDP.

First, consider that by the construction of the product MDP, Definition 7, preference graphs $\mathcal{G}$ and G are isomorphic, and thus, each $W_i \in \mathcal{W}$ is mapped to a single $F_i \in \mathbb{F}$, and vice versa. Let's define $F_i^+ = \bigcup_{F, F_i \rightsquigarrow F} F$ for each $F_i \in \mathbb{F}$. Given that $\mathcal{G}$ and G are isomorphic, $W_i \rightsquigarrow W_j$ if and only if $F_i \rightsquigarrow F_j$ for all $i, j \in \{1, 2, \cdots, N\}$. This combined with that $Z_i = \bigcup_{W \in \{W_i\}^{\uparrow}} W$ for $i \in \{1, \cdots, N\}$ by Definition 8, implies that for each i,

$$\mathbf{V}_i^{\mu} = \mathbf{Pr}^{\mu}(\text{reach}(Z_i), \mathcal{M}) = \mathbf{Pr}^{\pi}(\text{reach}(F_i^+), M\}. \quad (5)$$

Next, for each $w, w' \in \Sigma^*$ such that $\delta(\iota, w) = \delta(\iota, w')$, it holds that $\{w\}^{\uparrow} = \{w'\}^{\uparrow}$. Given this and Lemma 1, for each F_i and $w \in \Sigma^*$ such that $\delta(\iota, w) \in F_i$,

$$\mathbf{Pr}^{\pi}(\text{reach}(F_i^+), M) = \mathbf{Pr}^{\pi}(\{w\}^{\uparrow}). \quad (6)$$

Finally, given that μ is a Pareto optimal policy, by Definition 9 and Definition 10, it means there exists no policy μ' such that $\mathbf{V}_i^{\mu'} \geq \mathbf{V}_i^{\mu}$ for all integers $1 \leq i \leq n$ and $\mathbf{V}_j^{\mu'} > \mathbf{V}_j^{\mu}$ for some integer $1 \leq j \leq n$. This, by (5) and (6) and that the set of randomized, memoryless policies suffices for the Pareto front of $\mathcal{M}$, means there exists no policy $\pi' \in \Pi_{prop}^{M}$ such that $\mathbf{Pr}^{\pi'}(\{w\}^{\uparrow}) \geq \mathbf{Pr}^{\pi}(\{w\}^{\uparrow})$ for every $w \in \Sigma^*$ and $\mathbf{Pr}^{\pi'}(\{w'\}^{\uparrow}) > \mathbf{Pr}^{\pi}(\{w'\}^{\uparrow})$ for some $w' \in \Sigma^*$. This, by Definition 4 and Definition 5, means that π is weak-stochastic nondominated. □

Now one can use any existing methods to compute a set of Pareto optimal policies for $\mathcal{P}$. For a survey of those methods, see [13]. Note that computing the set of all Pareto optimal policies is generally infeasible, and thus, one needs to compute only a subset of them or to approximate them.

V. Case Study: Garden

In this section, we present the results from the planning algorithm for the running example in Figure 1 .

In the garden, the actions of the robot are N, S, E, W—for receptively moving to the cell in the North, South, East, and West side of the current cell—and T for staying in the current cell. The bee robot initially has a full charge, and using that charge it can fly only 12 time steps.

Uncertain environment: A bird roams about the south east part of the garden, colored yellow in the figure. When the bird and the bee are within the same cell, the bee needs to stop flying and hide in its current location until the bird goes away. The motion of the bird is given by a Markov chain. Besides the stochastic movement of the bird, the weather is also stochastic and affects the robot's planning. The robot cannot pollinate a flower while raining. We assume when the robot starts its task, at the leftmost cell at the bottom row, it is not raining and the probability that it will rain in the next step is 0.2. This probability increases for the consecutive steps each time by 0.2 until the rain starts. Once the rain started, the probability for the rain to stop in the five following time steps will respectively be 0.2, 0.4, 0.6, 0.8, and 1.0, assuming the rain has not already stopped at any of those time steps.

We implemented this case study in Python and considered two variants of it, one without stochasticity in the robot's dynamics, and one with stochasticity. In the former case, when the robot decides to perform an action to move to a neighboring cell, its actuators will guarantee with full certainty that the robot will move to that cell after performing the action. In the later case, the probability that the robot reaches the intended cell is 0.7, and for each of the unintended directions except the opposite direction, the probably that the robot's actuators move the robot to that unintended direction is 0.1. If the robot hits the boundary, it stays in its current cell.

All the experiments were performed on a Windows 11 installed on a device with a core i7, 2.80GHz CPU and a 16GB memory.

A. Deterministic Robot in the Uncertain Environment

The MDP for this case has $10,460$ states and $280,643$ transitions (its transition function has $280,643$ entries with non-zero probabilities). It took 47.38 seconds for our program to construct the MDP. The product MDP had $36,649$ states and $946,467$ transitions. The construction time for the product MDP was 408.87 seconds.

Given the preference described in Fig. 2, we employ linear scalarization methods to solve the MOMDP.

Specifically, given a weight vector $\mathbf{w} = [w_1, w_2, w_3, w_4]$, we compute the weak-stochastic nondominated policy $\mu_{\mathbf{w}}$, by first setting $V_{\mathbf{w}}(x) = \sum_{i:x \in Z_i} w_i$ for each goal state $x \in X_G$, and then by solving the following Bellman equation for the values of the non-goal states

$$V_{\mathbf{w}}(x) = \max_{a \in A_x} \sum_{x' \in X} \mathbf{T}(x, a, x') V_{\mathbf{w}}(x'), \forall x \in X \setminus X_G. \quad (7)$$

The policy is recovered from $V_{\mathbf{w}}(x)$ as

$$\mu_{\mathbf{w}}(x) = \operatorname*{argmax}_{a \in A_x} \sum_{x' \in X} \mathbf{T}(x, a, x') V_{\mathbf{w}}(x'), \forall x \in X \setminus X_G. \quad (8)$$

We randomly generated 100 weight vectors and used each one of them to compute a Pareto optimal policy for the MOMDP. The computed Pareto-optimal policies in the MOMDP yield a set of 100 weak-stochastic nondominated policies. From the result, it is noted that none of those computed polices were weak-stochastic dominated by the other polices. This is expected due to Theorem 1. Table I shows 10 out of those 100 weight vectors along with the value vectors of the polices computed for those weight vectors and the corresponding probabilities those polices assign to the four preferences p_1 through p_4.

	Weight Vector	Value Vector	Prob. of individual outcomes
1	[0.50, 0.17, 0.21, 0.12]	[0.24, 0.25, 0.98, 1.0]	[0.24, 0.01, 0.74, 0.01]
2	[0.08, 0.46, 0.38, 0.08]	[0.24, 0.42, 0.80, 1.0]	[0.24, 0.18, 0.56, 0.02]
3	[0.73, 0.13, 0.13, 0.01]	[0.24, 0.32, 0.91, 1.0]	[0.24, 0.08, 0.67, 0.01]
4	[0.67, 0.24, 0.02, 0.07]	[0.19, 0.63, 0.51, 1.0]	[0.19, 0.44, 0.32, 0.05]
5	[0.16, 0.11, 0.04, 0.69]	[0.15, 0.71, 0.42, 1.0]	[0.15, 0.56, 0.27, 0.02]
6	[0.26, 0.16, 0.03, 0.55]	[0.15, 0.72, 0.40, 1.0]	[0.15, 0.57, 0.25, 0.03]
7	[0.24, 0.46 0.26, 0.04]	[0.17, 0.64, 0.53, 1.0]	[0.17, 0.47, 0.36, 0.00]
8	[0.22, 0.28, 0.13, 0.37]	[0.15, 0.73, 0.40, 1.0]	[0.15, 0.58, 0.25, 0.02]
9	[0.07, 0.65, 0.04, 0.25]	[0.00, 1.00, 0.00, 1.0]	[0.00, 1.00, 0.00, 0.00]
10	[0.18, 0.08, 0.01, 0.73]	[0.18, 0.63, 0.51, 1.0]	[0.18, 0.45, 0.33, 0.04]

TABLE I: Ten weak-stochastic nondominant polices computed by our algorithm for the Garden case study.

For each policy, the last column shows probability vector indicating the probability distribution over individual outcomes $p_1, \ldots, p_4$ (in this order) given the computed policy. The third column shows the multi-objective value vector of each computed policy. It is noted that none of those value vectors dominates any other value vector.

Rows 1 and 3 of this table show that even if the weight assigned to the most preferred outcome, p_1, is significantly higher than the weights assigned to the other preferences, the probability that p_1 to be satisfied is still less than 0.25. This is justified by the fact that the robot's battery capacity supports the robot for only 12 time steps and thus to achieve p_1, the robot must not be stopped by the bird nor there should be raining when it reaches a cell to do pollination. The probability to satisfy these conditions given the environment dynamics is less than 0.25. The probability of p_4 to be satisfied in any entry of this table is less than 0.05. This is because p_4 has the lowest priority, and any policy would prefer to satisfy other preferences who are assigned higher priorities. Although the objectives $\{p_1, p_2\}$ and $\{p_1, p_3\}$ in the first and the third rows are treated almost equally by the weight vector in terms of importance, the probability that the later to be satisfied is significantly bigger than the probability of the former to be satisfied. This is because the objective $\{p_1\}$ contains the preference with the highest priority and that those two rows assign a very high weight to this objective, forcing the policy to try to satisfy p_1. Further, by attempting to perform p_1, the robot has the chance to accomplish p_3 within the same attempt, albeit if it fails to accomplish p_1. More precisely, if in attempting to perform the task p_1—first tulips and then at least one out of daisies and orchids—the robot succeeds to pollinate the tulips but fails to pollinate the daisies and orchids, then it has already accomplished p_3, even though it has failed in accomplishing what it was aiming for—p_1.

B. Stochasticity in the Robot's Actions

The MDP for this variant has the same number of states, $10,460$, but it has more transitions, $779,396$, which is due to the stochasticity in robot's dynamics. The MDP construction time for this case was 279.85 seconds and it took $2,129.02$ seconds to make the product MDP. We again computed 100 weak-stochastic nondominated policies.

Due to the stochasticity in the robot's dynamic, we expect the policy computed for a specific weight vector to be less "attractive" than a policy computed for the same weight vector of the previous variant. We compare those two polices for the weight vector $[0.25, 0.25, 0.25, 0.25]$. The probabilities of the preferences to be satisfied for the variant without stochasticity were $[p_1 : 0.24, p_2 : 0.05, p_3 : 0.70, p_4 : 0.01]$, while those probabilities for the variant with stochasticity were $[p_1 : 0.01, p_2 : 0.82, p_3 : 0.12, p_4 : 0.05]$. While the former policy yields a higher probability of achieving the most preferred temporal goal—p_1, the latter policy puts most of its efforts to satisfy p_2.

VI. Conclusions and Future Work

In this paper, we proposed a finite automaton for specifying user preferences over temporal goals, formulated and solved a preference-based temporal planning in stochastic systems. The characteristic that distinguishes our work from prior work on temporal logic planning is that our formulation consider the case where the user preferences may have incomparable outcomes, and this, introduces a problem in defining how to compare policies given their distributions over outcomes as well as a problem in designing planning algorithm to solve a preferred policy.

We use the notion of weak-stochastic ordering to rank different policies. Future work may consider other stochastic ordering that are used for ranking probability measures given a partial order over outcomes. Another direction will be to extend this work to the planning with preference over temporal goals that are satisfied in infinite time, for instance, recurrent properties and other more general properties in temporal logic. For practical robotic applications, it would be interesting to design an interface that translates human language preference specifications or human supervisors' feedback to a computational model, such as a preference automaton or its variant, to facilitate human-on-the-loop planning.

References

[1] Robert J Aumann. Utility theory without the completeness axiom. *Econometrica: Journal of the Econometric Society*, pages 445–462, 1962.

[2] Jorge A Baier and Sheila A. McIlraith. Planning with Preferences. *AI Magazine*, 29(4):25, 2008.

[3] Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

[4] Krishnendu Chatterjee, Rupak Majumdar, and Thomas A Henzinger. Markov decision processes with multiple objectives. In *Annual symposium on theoretical aspects of computer science*, pages 325–336. Springer, 2006.

[5] Jie Fu. Probabilistic planning with preferences over temporal goals. In *2021 American Control Conference (ACC)*, pages 4854–4859. IEEE, 2021.

[6] Reid Hastie and Robyn M Dawes. *Rational choice in an uncertain world: The psychology of judgment and decision making*. Sage, 2010.

[7] John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. Introduction to automata theory, languages, and computation. *Acm Sigact News*, 32(1):60–65, 2001.

[8] Morteza Lahijanian and Marta Kwiatkowska. Specification revision for Markov decision processes with optimal trade-off. In *Proc. 55th Conference on Decision and Control (CDC'16)*, pages 7411–7418, 2016.

[9] Meilun Li, Andrea Turrini, Ernst Moritz Hahn, Zhikun She, and Lijun Zhang. Probabilistic preference planning problem for markov decision processes. *IEEE transactions on software engineering*, 2020.

[10] William A. Massey. Stochastic Orderings for Markov Processes on Partially Ordered Spaces. *Mathematics of Operations Research*, 12(2):350–367, 1987. Publisher: INFORMS.

[11] Noushin Mehdipour, Cristian-Ioan Vasile, and Calin Belta. Specifying User Preferences Using Weighted Signal Temporal Logic. *IEEE Control Systems Letters*, 5(6):2006–2011, December 2021.

[12] Hazhar Rahmani and Jason M O'Kane. What to do when you can't do it all: Temporal logic planning with soft temporal logic constraints. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6619–6626. IEEE, 2020.

[13] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

[14] Ganesh Ram Santhanam, Samik Basu, and Vasant Honavar. Representing and Reasoning with Qualitative Preferences: Tools and Applications. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(1):1–154, January 2016. ZSCC: 0000006 Publisher: Morgan & Claypool Publishers.

[15] Jana Tumova, Gavin C Hall, Sertac Karaman, Emilio Frazzoli, and Daniela Rus. Least-violating control strategy synthesis with safety rules. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 1–10. ACM, 2013.

[16] Tichakorn Wongpiromsarn, Konstantin Slutsky, Emilio Frazzoli, and Ufuk Topcu. Minimum-violation planning for autonomous systems: Theoretical and practical considerations. In *2021 American Control Conference*, 2021. submitted.