Preference-Based Planning in Stochastic Environments: From Partially-Ordered Temporal Goals to Most Preferred Policies

Hazhar Rahmani[1], Abhishek N. Kulkarni[2] and Jie Fu[1]

Abstract

Human preferences are not always represented via complete linear orders: It is natural to employ partially-ordered preferences for expressing incomparable outcomes. In this work, we consider decision-making and probabilistic planning in stochastic systems modeled as Markov decision processes (MDPs), given a partially ordered preference over a set of temporally extended goals. Specifically, each temporally extended goal is expressed using a formula in Linear Temporal Logic on Finite Traces (LTL$_f$). To plan with the partially ordered preference, we introduce order theory to map a preference over temporal goals to a preference over policies for the MDP. Accordingly, a most preferred policy under a stochastic ordering induces a stochastic nondominated probability distribution over the finite paths in the MDP. To synthesize a most preferred policy, our technical approach includes two key steps. In the first step, we develop a procedure to transform a partially ordered preference over temporal goals into a computational model, called preference automaton, which is a semi-automaton with a partial order over acceptance conditions. In the second step, we prove that finding a most preferred policy is equivalent to computing a Pareto-optimal policy in a multi-objective MDP that is constructed from the original MDP, the preference automaton, and the chosen stochastic ordering relation. Throughout the paper, we employ running examples to illustrate the proposed preference specification and solution approaches. We demonstrate the efficacy of our algorithm using these examples, providing detailed analysis, and then discuss several potential future directions.

1 Introduction

With the rise of artificial intelligence and foundational models, robotics and other autonomous systems are now designed to understand and respond to human commands in natural language, making human-robot interactions more intuitive and user-friendly. However, human commands or preferences are not always expressible by a complete linear order. 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, the quality of an apple to that of banana, are fundamentally incomparable since they lack a standard basis for comparison. These situations motivate the need for a procedure that translates human preferences into a computational model for autonomous agents and a planner that deals with partially ordered preferences in the presence of all uncertainties in its environment.

In this paper, we consider preference-based planning (PBP) in stochastic systems modeled as Markov decision processes (MDPs) with user preferences over temporally extended goals. Specifically, we express each temporally extended goal using a formula in Linear Temporal Logic on Finite Traces (LTL$_f$). For motivation, 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.

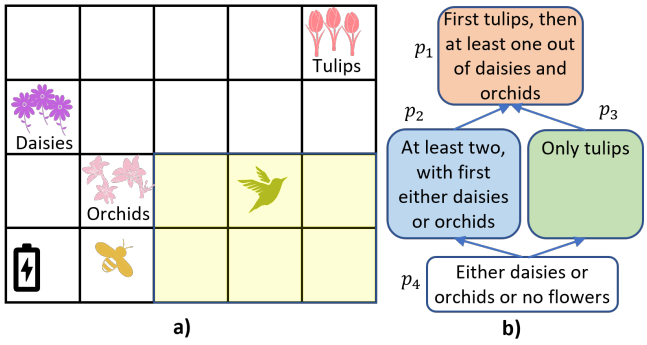


Figure 1. a) Bob's Garden. **b)** Bob's preferences on how the bee robot should perform the task of pollinating the flowers.

[1]Department of Electrical and Computer Engineering, University of Florida, USA
[2]Oden Institute for Computational Engineering and Sciences, University of Texas at Austin, TX, 78712, USA

Corresponding author:
Jie Fu, 1889 Museum Road, 5000 Malachowsky Hall, Gainesville, FL 32611, USA.
Email: {h.rahmani, fujie}@ufl.edu, abhishek.kulkarni@austin.utexas.edu.

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) pollinate either daisies or orchids or no flowers,

over which, his preference is shown in Figure 1b, using a preference graph, where the nodes represent the outcomes, and each directed edge is an improving flip (Santhanam et al. 2016). According to this graph, p_1 is the most preferred outcome, and p_4 is the least preferred, 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. Thus, we are interested to answer the following questions: Given all the uncertainties in the environment, how to compute a robot's policy that maximally satisfies Bob's preference? How to rank different policies, considering the fact that some outcomes are incomparable?

Preference-based planning (PBP) enables the system to decide which goals to satisfy when not all of them can be achieved (Hastie and Dawes 2010). Even though PBP has been studied since the early 1950's, most works on preference-based temporal planning (c.f. Baier and McIlraith (2008)) assume that all outcomes are pairwise comparable—that is, the preference relation is a *total* order. This assumption is often strong and, in many cases, unrealistic (Aumann 1962).

With the emergence of large language models translating human commands into temporal logic formulas (Chen et al. 2023; Cosler et al. 2023), it becomes natural to consider developing PBP algorithms with human preferences over temporal goals, which are commonly encountered in robotic planning problems. Setting aside natural language understanding, PBP has been well-studied for deterministic systems given both total and partial preferences. See the survey by Baier and McIlraith (2008). For preferences over temporal goals in deterministic systems, several works (Tumova et al. 2013; Wongpiromsarn et al. 2021; Rahmani and O'Kane 2020, 2019) proposed minimum-violation planning methods that decide which low-priority constraints should be violated. Amorese and Lahijanian (2023) formulate and solve a two-objective optimal planning problem where one objective is to minimize the total cost of a plan, while the other aims to optimize the costs of individual temporal goals ordered by the user preference.

Mehdipour et al. (2021) associate weights with Boolean and temporal operators in signal temporal logic to specify the importance of satisfying the sub-formulas and priority in the timing of satisfaction. This reduces the PBP problem to that of maximizing the weighted satisfaction in deterministic dynamical systems. For planning under this new specification language, Cardona et al. (2023) propose an algorithm based on mixed linear integer programming. However, the solutions to PBP for deterministic systems cannot be applied to stochastic systems (such as MDPs/POMDPs). 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.

Preference-based planning for stochastic systems has been less studied until recently. Lahijanian and Kwiatkowska (2016) consider a problem that aims to revise a given specification to improve the probability of satisfaction of the specification. They develop an Markov Decision Process (MDP) planning algorithm that trades off minimizing the cost of revision and maximizing the probability of satisfying the revised formula. Cai et al. (2021) focus on planning with infeasible LTL specifications in MDPs. Their problem's objective is to synthesize a policy that, in decreasing order of importance, 1) provides a desired guarantee to satisfy the task, 2) satisfies the specifications as much as possible, and 3) minimizes the implementation cost of the plan. Li et al. (2020) solve a preference-based probabilistic planning problem by reducing it to a multi-objective model checking problem. Li et al. (2023) study a class of preferences over temporal goals constructed using prioritized conjunction and ordered disjunction and show that these formulas can be equivalently expressed by weighted automata. They then provide a probabilistic planning algorithm that maximizes the expected degree of satisfaction. However, all these works assume the preference relation to be *total*. To the best of our knowledge, only Fu (2021) and Kulkarni and Fu (2022) have studied probabilistic planning with incomplete preferences. Kulkarni and Fu (2022) focus on the qualitative version of the problem, synthesizing strategies that identify and exploit opportunities to improve the most preferred achievable outcome with either positive probability or probability one. This is achieved by reducing the problem to reactive synthesis (Manna and Pnueli 2012; Baier and Katoen 2008). In Fu (2021), the author introduced 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 rank the policies in the stochastic system with respect to the partial order of temporal goals and allows the time horizon to be finite, but unbounded.

Our contributions in this paper are four-fold. (1) We introduce a new computational model called *Preference Deterministic Finite Automaton (PDFA)*, which models a user's (possibly partially-ordered) preference over temporally extended goals. (2) We introduce an algorithm that translates a set of partially ordered LTL$_f$ formulas, each representing a temporal goal, to a PDFA. (3) We establish a connection between the PBP in stochastic systems and the notions of stochastic orders (Massey 1987). This connection allows us to rank policies given their induced probabilistic distribution over possible outcomes. Hence, it reduces probabilistic planning with partially-ordered preferences over temporal goals to computing the set of nondominated policies for a multi-objective MDP, constructed as a product of the MDP modeling the environment and the PDFA specifying the user preference over the temporal goals. (4) We employ the property of weak-stochastic nondominated policies to design multiple objective functions 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.

The paper is organized as follows. In Section 2, we present preliminaries and our problem definition. In Section 3, we introduce Preference Deterministic Finite Automaton (PDFA), and in Section 4, we present our algorithm for converting a preference model of a set of LTL$_f$ formulas into a PDFA. In Section 5, we present our algorithm for computing a nondominated a policy, given the PDFA specifying the user's preference over temporal goals. In Section 6, we present a case study and our detailed analysis.

We presented a preliminary version of this paper at the 2023 IEEE International Conference on Robotics and Automation (Rahmani et al. 2023). In addition to revisions made throughout the paper, we have included several new results: (1) Our preliminary version assumed the PDFA is given by the user, but in this version we assume the user's preference is specified using a partially ordered set of LTL$_f$ formulas, and develop an algorithm to translate the partially ordered set of LTL$_f$ formulas into a PDFA; (2) the preliminary version considered only the notion of weak-stohastic ordering for comparing policies, but in this version we added two additional notions of stochastic ordering, strong-stochastic ordering and weak*-stochastic ordering; and (3) we extended our experiment to include results for the new additional stochastic orderings and discuss how different stochastic orders may affect the policy choices.

2 Definitions

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

2.1 The System and its Policy

We model the system using a variant of MDP.

Definition 1. Terminating Labeled Markov Decision Process (TLMDP). *A 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., $L(s) = \epsilon$ if and only if $s = s_\perp$.*

Though this definition assumes a single sink state, we do not lose generality, as one can always convert an MDP with multiple sink state into an equivalent MDP with a single sink state by keeping only a single sink state and redirecting all the transitions to other sink states 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 $0 \le i \le k$, 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 the distribution $\mathbf{P}(\cdot \mid s_i, a_i)$. 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 $\pi : \mathscr{D} \to \mathscr{C}$ with $\mathscr{D} \in \{S, S^*\}$ and $\mathscr{C} \in \{A, \mathcal{D}(A)\}$, and it is called *memoryless* if $\mathscr{D} = S$; *finite-memory* if $\mathscr{D} = S^*$; *deterministic* if $\mathscr{C} = A$, and *randomized* if $\mathscr{C} = \mathcal{D}(A)$.

In a terminating MDP, a policy is *proper* if it guarantees that the termination state $s_\perp$ will be reached with probability one (Bertsekas and Tsitsiklis 1991). The set of all randomized, finite-memory, proper polices for M is denoted Π^M_{prop}. In this paper, we consider only the TLMDPs for which all the policies are proper. In other words, the system always terminates after a finite time. This restriction is due to that (1) many applications require the robot to finish its execution in a finite time, and 2) the preference specification, defined next, is restricted to a partially ordered set of finite traces.

2.2 Specifying the Temporal Goals

The temporal goals in the MDP are specified formally using the following language.

Definition 2. Syntax of Linear Temporal Logic on Finite Traces (LTL$_f$) (De Giacomo and Vardi 2013). *Given a finite set $\mathcal{AP}$ of atomic propositions, a formula in LTL$_f$ over $\mathcal{AP}$ is generated by the following grammar:*

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc \varphi \mid \varphi \,\mathsf{U}\, \varphi,$$

where $p \in \mathcal{AP}$ is an atomic proposition, $\neg$ and $\wedge$ are the standard Boolean operators negation and conjunction, respectively, and $\bigcirc$ and U are temporal operators "Next" and "Until", respectively.

The temporal operators are interpreted over sequences of time instants. The formula $\bigcirc\varphi$ means at the next time instant, φ holds true. Formula $\varphi_1 \,\mathsf{U}\, \varphi_2$ holds true at the current time instant if there exists a future time instant at which φ_2 holds true and for all time instants from the current time until that future time, φ_1 holds true. An additional temporal operator "Eventually" ($\Diamond$) is defined as $\Diamond\varphi := \mathsf{true}\,\mathsf{U}\,\varphi$. Formula $\Diamond\varphi$ means there exists a future time instant at which φ holds true. The dual of Eventually operator is the "Always" ($\Box$). It is defined as $\Box\varphi := \neg\Diamond\neg\varphi$. Formula $\Box\varphi$ means φ holds true at the current instant and all future instants. For formal semantics of LTL$_f$, see De Giacomo and Vardi (2013).

Example 1. *For the example in Figure 1, we set $\mathcal{AP} = \{o, d, t\}$, in which o means orchids are being pollinated, d means daisies are being pollinated, and t means tulips are being pollinated. The temporal goals p_1 through p_4 in*

the example in that figure are expressed using the following LTL$_f$ formulas:

(p_1) pollinate tulips first, then at least one out of daisies and orchids;

$$(\neg d \wedge \neg o)\,\mathsf{U}\,(t \wedge \bigcirc \Diamond (d \vee o)),$$

(p_2) pollinate two types of flowers, with the first being either daisies or orchids;

$$\neg t\,\mathsf{U}\,((o \wedge \bigcirc \Diamond (d \vee t)) \vee (d \wedge \bigcirc \Diamond (o \vee t))),$$

(p_3) pollinate only tulips;

$$(\neg d \wedge \neg o)\,\mathsf{U}\,(t \wedge \Box (\neg d \wedge \neg o)), \text{ and}$$

(p_4) pollinate either daisies or orchids or no flowers;

$$\Box (\neg d \wedge \neg o \wedge \neg t) \vee (\Diamond o \wedge \Box (\neg d \wedge \neg t)) \vee (\Diamond d \wedge \Box (\neg o \wedge \neg t)).$$

Given an LTL$_f$ formula φ, the words over the alphabet $2^{\mathcal{AP}}$ that satisfy φ, constitute the language of φ, which is denoted $\mathcal{L}(\varphi)$. In the following context, we assume $\Sigma := 2^{\mathcal{AP}}$.

The language of LTL$_f$ formula φ can be represented by the set of words accepted by a finite automaton defined as follows:

Definition 3. Deterministic Finite Automaton (DFA). *A DFA is a tuple $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F\rangle$ with a finite set of states Q, a finite alphabet Σ, a deterministic transition function $\delta : Q \times \Sigma \to Q$, an initial state $q_0 \in Q$, and a set of accepting (final) states $F \subseteq Q$. For each state $q \in Q$ and letter $\sigma \in \Sigma$, $\delta(q, \sigma) = q'$ is the state reached upon reading input σ from state q.*

Slightly abusing the notion, we define the extended transition function $\delta : Q \times \Sigma^* \to Q$ in the usual manner: $\delta(q, \sigma w) = \delta(\delta(q, \sigma), w)$ for a given $\sigma \in \Sigma$ and $w \in \Sigma^*$, and $\delta(q, \epsilon) = q$. A word $w \in \Sigma^*$ is *accepted* by the DFA if and only if $\delta(q, w) \in F$. The language of $\mathcal{A}$, denoted $\mathcal{L}(\mathcal{A})$, is set the of all words accepted by the DFA, i.e., $\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^* \mid \delta(q, w) \in F\}$.

For each LTL$_f$ formula φ, there exists a DFA $\mathcal{A}_\varphi$ such that $\mathcal{L}(\varphi) = \mathcal{L}(\mathcal{A}_\varphi)$. Therefore, we can encode each LTL$_f$ formula using a DFA (De Giacomo and Vardi 2013).

2.3 Rank the Policies

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

Definition 4. *A preference model is a tuple $\langle U, \succeq\rangle$ where U is a countable set of outcomes and $\succeq$ is* preorder*–a reflexive and transitive binary relation–on U.*

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 5. *Given a preference model $\langle U, \succeq\rangle$, let X be a subset of U, the* upper closure of X *is defined by*

$$X^{\uparrow} = \{y \mid y \succeq x \text{ for some } x \in X\}, \text{ and}$$

the lower closure of X *is defined by*

$$X^{\downarrow} = \{y \mid y \preceq x \text{ for some } x \in X\}.$$

A set X is called an increasing set *if $X = X^{\uparrow}$.*

Massey (1987) introduced three different stochastic orderings, called, *strong*, *weak*, and *weak** orderings. The three stochastic orderings differ in how they determine a *family of subsets* of U.

Definition 6. *(Massey 1987) Let $\mathfrak{E}_{st}(U)$, $\mathfrak{E}_{wk}(U)$, and $\mathfrak{E}_{wk*}(U)$ denote the strong, weak, and weak* orderings, respectively. It is defined that*

$$\mathfrak{E}_{st}(U) = \{\text{ all increasing sets in } 2^U\},$$
$$\mathfrak{E}_{wk}(U) = \{\{x\}^{\uparrow} \mid x \in U\} \cup \{U, \emptyset\}, \text{ and}$$
$$\mathfrak{E}_{wk*}(U) = \{E \setminus \{x\}^{\downarrow} \mid x \in U\} \cup \{U, \emptyset\}.$$

These stochastic orderings allow us to rank probability measures according to the partially ordered set U. Let $\mathfrak{E}(U)$ be *a family of subsets* of U that includes U itself and the empty set $\emptyset$. That is, $\mathfrak{E} \in \{\mathfrak{E}_{st}, \mathfrak{E}_{wk}, \mathfrak{E}_{wk*}\}$. Let P_1 and P_2 be two probability measures on U. We denote $P_2 \geq_{\mathfrak{E}} P_1$ whenever $P_2(X) \geq_{\mathfrak{E}} P_1(X)$ for all subsets $X \in \mathfrak{E}(U)$. It is proven that if the partial order U is a total order, then the three stochastic orderings $\mathfrak{E}_{st}(U)$, $\mathfrak{E}_{wk}(U)$, and $\mathfrak{E}_{wk*}(U)$ are equivalent (see Proposition 2.5 of Massey (1987)). However, for a partial order, the three stochastic orderings may differ.

To illustrate, consider the following example.

Example 2. *Let $U = \{a, b, c, d\}$ and $\succeq = \{(a, b), (b, d), (c, d), (a, c), (a, d)\} \cup I_U$, where I_U is the identity relation on U and that $(x, y) \in \succeq$ if and only if $x \succeq y$. Also, consider probability measures P_1, P_2, and P_3 where $P_1(a) = 0.5, P_1(b) = 0.3, P_1(c) = 0.2$, $P_2(b) = 0.5, P_2(c) = 0.3, P_2(d) = 0.2$, and $P_3(a) = 0.3, P_3(b) = 0.2, P_3(d) = 0.5$.*

We have

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

Accordingly,

$$[P_1[X]]_{X \in \mathfrak{E}_{st}(U)} = [0.5, 0.8, 0.7, 1, 1, 0],$$
$$[P_2[X]]_{X \in \mathfrak{E}_{st}(U)} = [0, 0.5, 0.3, 1, 1, 0], \text{ and}$$
$$[P_3[X]]_{X \in \mathfrak{E}_{st}(U)} = [0.3, 0.5, 0.3, 0.5, 1, 0].$$

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

Also, we have

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

Accordingly,

$$[P_1[X]]_{X \in \mathfrak{E}_{wk}(U)} = [0.5, 0.8, 0.7, 1, 0],$$
$$[P_2[X]]_{X \in \mathfrak{E}_{wk}(U)} = [0, 0.5, 0.3, 1, 0], \text{ and}$$

$$[P_3[X]]_{X\in\mathfrak{E}_{wk}(U)} = [0.3, 0.5, 0.3, 1, 0].$$

Thus, $P_1 >_{\mathfrak{E}_{wk}} P_2$, $P_1 >_{\mathfrak{E}_{wk}} P_3$, *and* $P_3 >_{\mathfrak{E}_{wk}} P_2$.

Also, we have

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

Accordingly,

$$[P_1[X]]_{X\in\mathfrak{E}_{wk*}(U)} = [0.8, 0.7, 1, 1, 0],$$
$$[P_2[X]]_{X\in\mathfrak{E}_{wk*}(U)} = [0.5, 0.3, 0.8, 1, 0], \text{ and}$$
$$[P_3[X]]_{X\in\mathfrak{E}_{wk*}(U)} = [0.5, 0.3, 0.5, 1, 0].$$

Thus, $P_1 >_{\mathfrak{E}_{wk*}} P_2$, $P_1 >_{\mathfrak{E}_{wk*}} P_3$, *and* $P_2 >_{\mathfrak{E}_{wk*}} P_3$.

In our context, because we are interested in sequential decision-making and planning problems with a finite time termination, the set U is selected to be Σ^*, or the set of finite traces generated by the system and its labeling function. Based on the ranking of probability measures induced by each one of the stochastic orderings 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 $\rho \in \text{Paths}_\perp(M)$, $\mathbf{Pr}^\pi(\rho)$ 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 $\text{trace}(\rho) = L(s_0)L(s_1)\ldots$, 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. Formally,

$$\mathbf{Pr}^\pi(w) = \sum_{\rho\in\text{Paths}_\perp(M):L(\rho)=w} \mathbf{Pr}^\pi(\rho).$$

Additionally, for a subset $X \subseteq \Sigma^*$, $\mathbf{Pr}^\pi(X)$ is the probability of the words generated by π to be within X. Formally,

$$\mathbf{Pr}^\pi(X) = \sum_{w\in X} \mathbf{Pr}^\pi(w).$$

Definition 7. *Let* $\mathfrak{E} \in \{\mathfrak{E}_{st}, \mathfrak{E}_{wk}, \mathfrak{E}_{wk*}\}$ *be a stochastic ordering and* $\langle U := \Sigma^*, \succeq\rangle$ *be a preference model. Given two proper policies* π *and* π' *for the terminating labeled MDP* M, π $\mathfrak{E}$-stochastic dominates π', *denoted* $\pi >_{\mathfrak{E}} \pi'$, *if for each subset* $X \in \mathfrak{E}(U)$, *it holds that* $\mathbf{Pr}^\pi(X) \geq \mathbf{Pr}^{\pi'}(X)$, *and there exists a subset* $Y \in \mathfrak{E}(U)$ *such that* $\mathbf{Pr}^\pi(Y) > \mathbf{Pr}^{\pi'}(Y)$.

This definition is used to introduce the following notion.

Definition 8. *A proper policy* $\pi \in \Pi^M_{prop}$ *is* $\mathfrak{E}$-stochastic nondominated *if there* does not exist *any policy* $\pi' \in \Pi^M_{prop}$ *such that* $\pi' >_{\mathfrak{E}} \pi$.

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

We aim to solve the following planning problem:

Problem 1. *Given a terminating labeled MDP* $M = \langle S, A := \bigcup_{s\in S} A_s, \mathbf{P}, s_0, s_\perp, \mathcal{AP}, L\rangle$, *a preference model* $\langle\Sigma^*, \succeq\rangle$, *and a stochastic ordering* $\mathfrak{E} \in \{\mathfrak{E}_{st}, \mathfrak{E}_{wk}, \mathfrak{E}_{wk*}\}$, *compute a proper policy that is* $\mathfrak{E}-$*stochastic nondominated.*

3 Modeling Preference over LTL$_f$ Goals

In this section, we consider the case when the user defines their preference over temporal goals.

The user specifies the temporal goals using LTL$_f$ formulas, one formula for each goal, and then expresses their preference over these goals using a preference model over the set of these formulas.

Definition 9. *An* LTL$_f$ preference model *is a preference model* $\langle\Phi, \trianglerighteq\rangle$ *in which* $\Phi = \{\varphi_1, \ldots, \varphi_N\}$ *is a finite set of* distinct *LTL$_f$ formulas over a set of atomic propositions* $\mathcal{AP}$ *and* $\trianglerighteq$ *is a partial order—a reflexive, transitive, and an antisymmetric—relation on* Φ.

Two LTL$_f$ formulas φ and φ' are distinct if $\mathcal{L}(\varphi) \neq \mathcal{L}(\varphi')$, where $\mathcal{L}(\varphi)$ is the language of the formula, i.e., the set of words satisfying the formula.

Assumption 1. *We assume that* $\bigcup_{1\leq i\leq N} \mathcal{L}(\varphi_i) = \Sigma^*$, *meaning that for each word* $w \in \Sigma^*$, *there is at least one* $\varphi \in \Phi$ *such that* $w \in \mathcal{L}(\varphi)$. *Note that if* $\bigcup_{1\leq i\leq N} \mathcal{L}(\varphi_i) \subset \Sigma^*$, *then the assumption will hold by adding the formula* $\varphi = \bigwedge_{1\leq i\leq N} \neg\varphi_i$ *to* Φ.

Assumption 2. *The preference model* $\langle\Phi, \trianglerighteq\rangle$ *is a partial order relation over* Φ, *which means the following properties are satisfied:*

- *Reflexive:* $\varphi \trianglerighteq \varphi$ *for all* $\varphi \in \Phi$.
- *Antisymmetric:* $\varphi \trianglerighteq \varphi'$ *and* $\varphi' \trianglerighteq \varphi$ *implies* $\varphi = \varphi'$.
- *Transitive:* $\varphi_1 \trianglerighteq \varphi_2$ *and* $\varphi_2 \trianglerighteq \varphi_3$ *implies* $\varphi_1 \trianglerighteq \varphi_3$.

Remark 1. *Note that any preference model* $\langle\Phi, \trianglerighteq\rangle$ *in which* $\trianglerighteq$ *is a preorder, i.e., a partial order without the requirement of being antisymmetric, can be converted into a preference model* $\langle\Phi', \trianglerighteq'\rangle$ *in which* $\trianglerighteq'$ *is a partial order. The idea is to iteratively refine the preference model until the resulting model has no pair of indifferent formulas. In each step, two indifferent formulas* φ_i *and* φ_j *are replaced by their disjunction* $\varphi_i \vee \varphi_j$, *after which the preference relation is altered accordingly.*

The model $\langle\Phi, \trianglerighteq\rangle$ is a combinative preference model, as opposed to an exclusionary one. This is because we do not assert the exclusivity condition $\varphi_i \wedge \varphi_j = \mathsf{false}$. This allows us to represent a preference such as "Visiting A and B is preferred to visiting A," ($\Diamond A \wedge \Diamond B \trianglerighteq \Diamond A$) where the less preferred outcome must be satisfied first in order to satisfy the more preferred outcome. In literature, it is common to study exclusionary preference models (see Baier and McIlraith (2008); Bienvenu et al. (2011) and the references within) because of their simplicity Hansson and Grüne-Yanoff (2022). However, we focus on planning with combinative preferences since they are more expressive than the exclusionary ones (Hansson 2001). In fact, every exclusionary preference model can be transformed into a combinative one, but the opposite is not true.

When a combinative preference model is interpreted over finite words, the agent needs a way to compare the sets of temporal logic objectives satisfied by two words. For instance, consider the preference that "Visiting A and B is preferred to visiting A," and let $w_1 = \emptyset\{A\}\emptyset\{B\}$ and $w_2 = \emptyset\{A\}\emptyset$ be two finite words. Note that w_1 has both A

and B evaluated true, each at some point in time, and $w_2 = \emptyset\{A\}\emptyset$ only has A evaluated true. Therefore, $w_1 \models \varphi_1 \wedge \varphi_2$, whereas w_2 satisfies only φ_2. To determine the preference between w_1 and w_2, the agent compares the set $\{\varphi_1, \varphi_2\}$ with $\{\varphi_2\}$ to conclude that w_1 is preferred over w_2. However, suppose the given preference is that "visiting A is preferred over visiting B," i.e., $(\Diamond A \trianglerighteq \Diamond B)$. Then the two words w_1 and w_2 are indifferent since both satisfy the more preferred objective $\Diamond A$. To formalize this notion, we define the notion of most-preferred outcomes.

Given a non-empty subset $\mathbb{X} \subseteq \Phi$, let $\mathsf{MP}(\mathbb{X}) \triangleq \{\varphi \in \mathbb{X} \mid \nexists \varphi' \in \mathbb{X} : \varphi' \trianglerighteq \varphi\}$ denote the set of most-preferred outcomes in $\mathbb{X}$.

Definition 10. *Given an LTL$_f$ preference model $\langle \Phi, \trianglerighteq \rangle$ and a finite word $w \in \Sigma^*$, the set of most-preferred formulas satisfied by w is given by* $\mathsf{MP}(w) := \mathsf{MP}(\{\varphi \in \Phi \mid w \models \varphi\})$.

By definition, there is no outcome included in $\mathsf{MP}(w)$ that is weakly preferred to any other outcome in $\mathsf{MP}(w)$. Thus, we have the following result.

Lemma 1. *For any word $w \in \Sigma^*$, formulas in $\mathsf{MP}(w)$ are incomparable to each other.*

Proof. By contradiction. Suppose that the set $\mathsf{MP}(w)$ contains two formulas φ_1 and φ_2 that are comparable. Then, since $\trianglerighteq$ is a partial order, one of the following cases must be true: 1) $\varphi_1 \triangleright \varphi_2$, or 2) $\varphi_2 \triangleright \varphi_1$. Consider the first case. By definition of MP operator, only φ_1 is included in $\mathsf{MP}(w)$. Similarly, in second case, only φ_2 is included in $\mathsf{MP}(w)$. This is a contradiction.

Now, we formally define the interpretation of $\langle \Phi, \trianglerighteq \rangle$ in terms of the preference relation it induces on Σ^*.

Definition 11. *An LTL$_f$ preference model $\langle \Phi, \trianglerighteq \rangle$ induces the preference model $\langle \Sigma^*, \succeq \rangle$ where for any $w_1, w_2 \in \Sigma^*$,*

- *$w_1 \succeq w_2$ if and only if for every formula $\varphi \in \mathsf{MP}(w_1)$, there exists a formula $\varphi' \in \mathsf{MP}(w_2)$ such that $\varphi \trianglerighteq \varphi'$,*
- *$w_1 \sim w_2$ if and only if $\mathsf{MP}(w_1) = \mathsf{MP}(w_2)$, and*
- *$w_1 \nparallel w_2$, otherwise.*

The following set of properties can be shown.

Lemma 2. *Letting $\langle \Sigma^*, \succeq \rangle$ be the preference model induced by $\langle \Phi, \trianglerighteq \rangle$, for any $w_1, w_2 \in \Sigma^*$, if $w_1 \succeq w_2$, then there does not exist a pair of outcomes $\varphi \in \mathsf{MP}(w_1)$ and $\varphi' \in \mathsf{MP}(w_2)$ such that $\varphi' \trianglerighteq \varphi$.*

Proof. By contradiction. Let $\mathsf{MP}(w_1) = \{\varphi_1, \ldots, \varphi_m\}$ and $\mathsf{MP}(w_2) = \{\psi_1, \ldots, \psi_t\}$. Suppose there exists $\psi \in \mathsf{MP}(w_2)$ such that $\psi \trianglerighteq \varphi$ for some $\varphi \in \mathsf{MP}(w_1)$. Given the assumption $w_1 \succeq w_2$, by Definition 11, there exists $\psi' \in \mathsf{MP}(w_2)$ such that $\varphi \trianglerighteq \psi'$. As a result, $\psi \trianglerighteq \varphi \trianglerighteq \psi'$, implying that $\psi \trianglerighteq \psi'$. This contradicts the result in Lemma 1 which imposes $\mathsf{MP}(w_2)$ to contain only incomparable formulas. Thus, the assumption that $\psi \trianglerighteq \varphi$ is contradicted.

Lemma 3. *If $w_1 \sim w_2$, then $\mathsf{MP}(w_1) = \mathsf{MP}(w_2)$.*

Algorithm 1 Construction of Preference Graph

```
1:  function PrefGraph(⟨Φ, ⊵⟩, ⟨Q, Σ, δ, ι⟩)
2:      Initialize 𝒲 = ∅, E = ∅, Z = ∅.
3:      for all (q, q′) ∈ Q × Q do
4:          if MP(q) = MP(q′) then
5:              Z = Z ∪ {(q, q′), (q′, q)}
6:          else
7:              Initialize D as an empty set of sets.
8:              for all φ ∈ MP(q) do
9:                  Add {φ′ ∈ MP(q′) | φ ⊵ φ′} to D.
10:             if ∅ ∉ D then
11:                 Z ← Z ∪ {(q′, q)}.
12:     𝒲 ← getSCC(⟨Q, Z⟩)
13:     for all W, W′ ∈ 𝒲 do
14:         if ∃q ∈ W, q′ ∈ W′ : (q, q′) ∈ Z then
15:             E ← E ∪ {(W, W′)}
16:     return G = ⟨𝒲, E⟩
```

Proof. By way of contradiction, suppose $\mathsf{MP}(w_1) \neq \mathsf{MP}(w_2)$. Without loss of generality, let $\varphi \in \mathsf{MP}(w_1) \setminus \mathsf{MP}(w_2)$. Given $w_1 \succeq w_2$ and $\varphi \in \mathsf{MP}(w_1)$, there must exist a formula $\psi \in \mathsf{MP}(w_2)$ such that $\varphi \trianglerighteq \psi$. Also, because $w_2 \succeq w_1$, there exists a formula $\phi \in \mathsf{MP}(w_1)$ such that $\psi \trianglerighteq \phi$. Due to the transitivity of $\trianglerighteq$, $\varphi \trianglerighteq \psi \trianglerighteq \phi$ and thus $\varphi \trianglerighteq \phi$ and that $\varphi, \phi \in \mathsf{MP}(w_1)$, contradicting Lemma 1. Since φ is chosen arbitrarily, witnessing this contradiction implies that $\mathsf{MP}(w_1) = \mathsf{MP}(w_2)$.

Lemma 4. *The preference model $\langle \Sigma^*, \succeq \rangle$ induced by $\langle \Phi, \trianglerighteq \rangle$ is a preorder.*

Proof. For any $w \in \Sigma^*$, $w \succeq w$ because for any $\varphi \in \mathsf{MP}(w)$, $\varphi \trianglerighteq \varphi$. Thus, $\langle \Sigma^*, \succeq \rangle$ is reflexive. For the transitivity, supposing $w_1 \trianglerighteq w_2$ and $w_2 \trianglerighteq w_3$, we need to show that $w_1 \trianglerighteq w_3$. Let for each $t \in \{1, 2, 3\}$, $\mathsf{MP}(w_t) = \{\varphi_{t,i} \mid i = 1, \ldots, n_t\}$ be the most prefered formulas satisfied by w_t. Given that $w_1 \succeq w_2$, for any $\varphi_{1,i} \in \mathsf{MP}(w_1)$, there exists $\varphi_{2,j} \in \mathsf{MP}(w_2)$ such that $\varphi_{1,i} \trianglerighteq \varphi_{2,j}$. Also, because $w_3 \succeq w_3$, for any such $\varphi_{2,j}$, there exists $\varphi_{3,k} \in \mathsf{MP}(w_3)$ such that $\varphi_{2,j} \trianglerighteq \varphi_{3,k}$. Using the transitivity property of $\trianglerighteq$, $\varphi_{1,i} \trianglerighteq \varphi_{3,k}$ holds. As a result, $w_1 \succeq w_3$.

Note that the preference relation $\langle \Sigma^*, \succeq \rangle$ needs not to be antisymmetric since there might exist two words $w_1 \neq w_2$ such that $\mathsf{MP}(w_1) = \mathsf{MP}(w_2)$. For example, with $\Phi = \{\Diamond a, \Diamond b\}$, consider two words $\emptyset\emptyset\{a\}\{b\}$ and $\emptyset\{a\}\emptyset\{b\}$. Since they both satisfy both $\Diamond a$ and $\Diamond b$, $w_1 \succeq w_2$ and $w_2 \succeq w_1$, while $w_1 \neq w_2$, showing an example where $\succeq$ is not antisymmetric.

4 Constructing a Computational Model for an LTL$_f$ Preference Model

In this section, we introduce a novel computational model called a Preference Deterministic Finite Automaton (PDFA), which encodes the preference model $\langle \Sigma^*, \succeq \rangle$ into an automaton. We present a procedure to construct a PDFA for a given preference model $\mathcal{P} = \langle \Phi, \trianglerighteq \rangle$ and prove its correctness with respect to the interpretation in Definition 11.

Definition 12. *A PDFA for an alphabet Σ is a tuple $\mathcal{A} = \langle Q, \Sigma, \delta, \iota, G := (\mathcal{W}, E)\rangle$ in which Q is a finite set of states; Σ is the alphabet; $\delta : Q \times \Sigma \to Q$ is a transition function; $\iota \in Q$ is the initial state; and $G = (\mathcal{W}, E)$ is a* preference graph *in which, $\mathcal{W} = \{W_1, W_2, \cdots, W_m\}$ is a partition of Q—i.e., $W \subseteq Q$ for each $W \in \mathcal{W}$, $W \cap W' = \emptyset$ for each distinct state subsets $W, W' \in \mathcal{W}$, and $\bigcup_{W \in \mathcal{W}} W = Q$; and $E \subseteq \mathcal{W} \times \mathcal{W}$ 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 12 augments a DFA with the preference graph G, instead of a set of accepting (final) states.

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

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

For an example, see Figure 2, which shows the PDFA for the preferences p_1 through p_4 in Figure 1.

Next, we describe the construction of PDFA given a preference model $\mathcal{P} = \langle \Phi, \trianglerighteq \rangle$. The construction involves two steps, namely, the construction of the underlying graph of PDFA and the construction of the preference graph.

Definition 13. *Given a preference model $\langle \Phi, \trianglerighteq \rangle$, for each formula $\varphi_i \in \Phi$, let $\mathcal{A}_i = \langle Q_i, \Sigma, \delta_i, \iota_i, F_i \rangle$ be the DFA representing the language of φ_i. The underlying automaton of the PDFA representing $\langle \Phi, \trianglerighteq \rangle$ is the tuple,*

$$\langle Q, \Sigma, \delta, \iota \rangle$$

in which $Q = Q_1 \times Q_2 \times \cdots Q_n$ is the set of states in PDFA; $\delta : Q \times \Sigma \to Q$ is a deterministic transition function where for each $\mathbf{q} = (q_1, q_2, \cdots, q_n) \in Q$ and $a \in \Sigma$, $\delta(\mathbf{q}, a) = (\delta_1(q_1, a), \delta_2(q_2, a), \cdots, \delta(q_n, a))$; and $\iota = (\iota_1, \iota_2, \cdots, \iota_n)$ is the initial state.

We represent each state in Q as a vector $\mathbf{q}$ and the i-th component of $\mathbf{q}$, denoted as $\mathbf{q}[i]$, is the state in Q_i.

Algorithm 1 describes a procedure to construct the preference graph. It uses the following definition that slightly abuses the notation $\mathsf{MP}(\cdot)$: For each product state $\mathbf{q}$, we define the set

$$\mathsf{MP}(\mathbf{q}) = \mathsf{MP}(\{\varphi_i \in \Phi \mid \mathbf{q}[i] \in F_i\})$$

In words, $\mathsf{MP}(\mathbf{q})$ is a set of most preferred formulas satisfied by any word that ends in $\mathbf{q}$.

Given the preference model $\langle \Phi, \trianglerighteq \rangle$ and the underlying automaton $\langle Q, \Sigma, \delta, \iota \rangle$ of the PDFA, lines 3–11 of Algorithm 1 construct a set Z of directed edges such that $(\mathbf{q}', \mathbf{q}) \in Z$ if and only if for every $\varphi \in \mathsf{MP}(\mathbf{q})$, there exists a formula $\varphi' \in \mathsf{MP}(\mathbf{q}')$ such that $\varphi \trianglerighteq \varphi'$. Lines 12–15 of Algorithm 1 shows how the set $\mathcal{W}$ and edges E of the preference graph are constructed. Using the set of directed edges Z, the algorithm computes first the set $\mathcal{W}$ as the set of strongly connected components of the graph given by state set Q and edges Z. Then, a directed edge from an SCC W to another SCC W' is added if there is a state $\mathbf{q} \in W$ and a state $\mathbf{q} \in W'$ such that $(\mathbf{q}, \mathbf{q}') \in Z$.

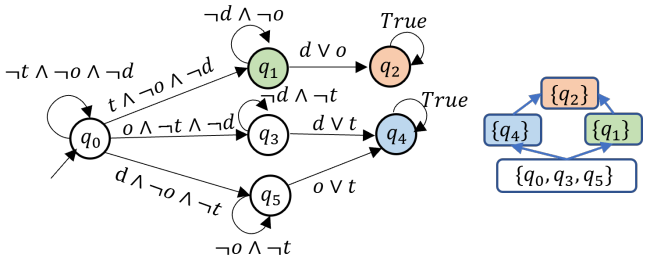


Figure 2. PDFA for the example in Figure 1. **Left)** The DFA structure of PDFA. **Right)** The preference graph of PDFA.

In Section 6, we provide a detailed explanation of the construction of the PDFA for the preferences in Figure 1, implemented through our algorithm.

We next show how the PDFA constructed using the product operation in Definition 13 and Algorithm 1 encodes the exact preference model $\langle \Phi, \trianglerighteq \rangle$.

Proposition 1. *Let $\mathcal{W}$ be the set of nodes constructed by Algorithm 1. For each $\mathbf{q}, \mathbf{q}' \in Q$ for which $\mathsf{MP}(\mathbf{q}) = \mathsf{MP}(\mathbf{q}')$, it holds that $\mathbf{q}$ and $\mathbf{q}'$ are included in the same node in $\mathcal{W}$.*

Proof. By the construction in Line 5 of Algorithm 1 and the definition of strongly connected components (Cormen et al. 2022).

Proposition 2. *If $(\mathbf{q}, \mathbf{q}') \notin Z$, then in graph $\langle Q, Z \rangle$ there is no directed path from $\mathbf{q}$ to $\mathbf{q}'$.*

Proof. For the sake of contradiction, suppose $(\mathbf{q}, \mathbf{q}') \notin Z$ but there is a directed path of length greater than 1 from $\mathbf{q}$ to $\mathbf{q}'$. Let this path be $\mathbf{q}_0 \to \mathbf{q}_1 \to \cdots \to \mathbf{q}_n$ where $\mathbf{q}_0 = \mathbf{q}$, $\mathbf{q}_n = \mathbf{q}'$, and $\mathbf{q}_1$ through $\mathbf{q}_{n-1}$ are intermediate states along the path. For $i = 0, \ldots, n$, let $X_i = \mathsf{MP}(\mathbf{q}_i)$. By the construction, for any $\varphi \in X_{i+1}$, there exists a formula $\psi \in X_i$ such that $\varphi \trianglerighteq \psi$. Applying the transitivity of the preference $\trianglerighteq$, it holds that for any $\varphi \in X_n = \mathsf{MP}(\mathbf{q}')$, there exists a formula $\psi \in X_0 = \mathsf{MP}(\mathbf{q})$ such that $\varphi \trianglerighteq \psi$. As a result, $(\mathbf{q}, \mathbf{q}') \in Z$, contradicting the assumption.

Proposition 3. *Set $\mathcal{W}$ constructed by Algorithm 1 partitions Q.*

Proof. This property automatically holds due to the property of strongly connected components (Cormen et al. 2022).

Theorem 1. *Let $\langle \Sigma^*, \succeq \rangle$ be the preference model induced by the semantics of $\langle \Phi, \trianglerighteq \rangle$ (Definition 11). Given the PDFA $\mathcal{A} = \langle Q, \Sigma, \delta, \iota, G \rangle$ constructed for the preference model $\langle \Phi, \trianglerighteq \rangle$ using Definition 13 and Algorithm 1, for any $w, w' \in \Sigma^*$ let $W, W' \in \mathcal{W}$ be the nodes such that $\delta(\iota, w) \in W$ and $\delta(\iota, w') \in W'$, the following statements are established:*

- *(Case 1) $W = W'$ if and only if $w' \sim w$.*
- *(Case 2) $W \neq W'$ and $W \to W'$ if and only if $w' \succeq w$ and $w \not\sim w'$.*

- *(Case 3)* $W \neq W'$ *and* $W' \to W$ *if and only if* $w \succeq w'$ *and* $w \not\sim w'$.
- *(Case 4)* $w \nparallel w'$, *otherwise.*

Proof. Let $\mathbf{q} = \delta(\iota, w)$ and $\mathbf{q}' = \delta(\iota, w')$. By construction of the function $\mathsf{MP}(\cdot)$ and the product operation in Definition 13, the following equation holds:

$$\begin{aligned}\mathsf{MP}(w) &= \mathsf{MP}(\{\varphi_i \mid \delta_i(\iota_i, w) \in F_i\}) \\ &= \mathsf{MP}(\{\varphi_i \mid \mathbf{q}[i] \in F_i\}) \\ &= \mathsf{MP}(\mathbf{q})\end{aligned}$$

Case 1: ($\Rightarrow$) If $W = W'$, then both $\mathbf{q}, \mathbf{q}' \in W$. This means that $\mathbf{q} \rightsquigarrow \mathbf{q}'$ and $\mathbf{q}' \rightsquigarrow \mathbf{q}$. By proposition 2, it is only possible that $(\mathbf{q}, \mathbf{q}') \in Z$ and $(\mathbf{q}', \mathbf{q}) \in Z$. $\mathsf{MP}(\mathbf{q}) = \mathsf{MP}(\mathbf{q}')$ can be derived due to the antisymmetric property in the partial order of $\langle \Phi, \trianglerighteq \rangle$.

($\Leftarrow$) If $w \sim w'$, then $\mathsf{MP}(\mathbf{q}) = \mathsf{MP}(\mathbf{q}')$ and therefore $W = W'$.

Case 2: ($\Rightarrow$) If $W \to W'$, then given the construction of the preference graph by lines 13-15 of Algorithm 1, there exist $q \in W$ and $q' \in W'$ such that $(q, q') \in Z$. Therefore, by construction in Algorithm 1, for any $\varphi' \in \mathsf{MP}(\mathbf{q}')$, there is a $\varphi \in \mathsf{MP}(\mathbf{q})$ such that $\varphi' \trianglerighteq \varphi$. Then, by Def. 11, $w' \succeq w$.

($\Leftarrow$) If $w' \succeq w$, then $(\mathbf{q}, \mathbf{q}') \in Z$. Because $W \neq W'$, then there is no path from $\mathbf{q}'$ to $\mathbf{q}$. As a result, it is not the case that $w \succeq w'$. Thus, $w \not\sim w'$.

Case 3: proof similar to the proof of Case 2.

Case 4 is a direct consequence from Cases 1, 2 and 3.

Using the computational model PDFA, we can directly compute the set $\{w\}^{\uparrow}$ for any $w \in \Sigma^*$.

Lemma 5. *For each word* $w \in \Sigma^*$, *if* $\delta(\iota, w) \in W$ *for some* $W \in \mathcal{W}$, *then the upper closure of* w *is*

$$\{w\}^{\uparrow} = \{w' \in \Sigma^* \mid \exists W' \in \mathcal{W}, \delta(\iota, w') \in W' \text{ and } W \rightsquigarrow W'\}, \quad (1)$$

and the lower closure of w *is*

$$\{w\}^{\downarrow} = \{w' \in \Sigma^* \mid \exists W' \in \mathcal{W}, \delta(\iota, w') \in \mathcal{W}' \text{ and } W' \rightsquigarrow W\} \quad (2)$$

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

Example 3. *Consider three LTL$_f$ formulas* $\varphi_1 = \Diamond a$, $\varphi_2 = \Diamond b$, *and* $\varphi_3 = \neg\Diamond a \wedge \neg\Diamond b$. *Also, assume* $\varphi_1 \triangleright \varphi_2$, $\varphi_1 \triangleright \varphi_3$, *and* $\varphi_2 \nparallel \varphi_3$. *The left column of Figure 3 shows for each of the three LTL$_f$ formula, a DFA that encodes that formula. The column in right shows the PDFA our algorithm constructs for these three formulas and the associated user preferences. In this PDFA, we have written in blue for each state* x, $\mathsf{Outcomes}(x)$, *the set of formulas satisfied when the word ends at state* x. *For each state* x, *we have also written in red,* $\mathsf{MP}(x)$*—the most preferred formulas among those formulas in* $\mathsf{Outcomes}(x)$. *Accordingly,* $\mathsf{Outcomes}(qpr) = \{\varphi_3\}$, $\mathsf{Outcomes}(q'pr') = \{\varphi_1\}$, $\mathsf{Outcomes}(qp'r') = \{\varphi_2\}$, *and* $\mathsf{Outcomes}(q'p'r') = \{\varphi_1, \varphi_2\}$. *Also,* $\mathsf{MP}(qpr) = \{\varphi_3\}$, $\mathsf{MP}(q'pr') = \{\varphi_1\}$, $\mathsf{MP}(qp'r') = \{\varphi_2\}$, *and* $\mathsf{MP}(q'p'r') = \{\varphi_1\}$. *Note that because* $\varphi_1 \triangleright \varphi_2$, $\mathsf{MP}(q'p'r') = \mathsf{MP}(\{\varphi_1, \varphi_2\}) = \{\varphi_1\}$. *Given that* $\mathsf{MP}(q'p'r') = \mathsf{MP}(q'pr')$, *states* $q'p'r'$ *and* $q'pr'$ *belong to the same vertex in the preference graph.*

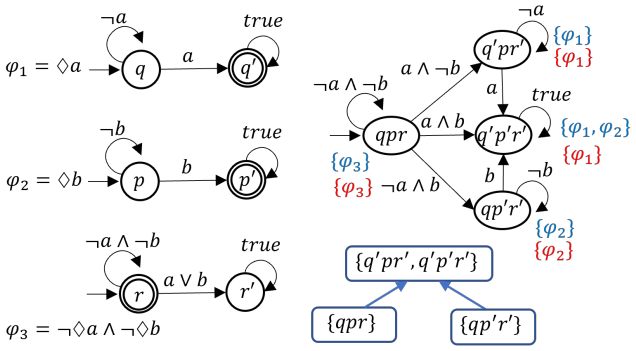


Figure 3. PDFA for the example in Figure 1. **Left)** Three DFAs for three LTL$_f$ formulas φ_1, φ_2, and φ_3, for which the user preference is: $\varphi_1 \triangleright \varphi_2$, $\varphi_1 \triangleright \varphi_3$, $\varphi_2 \nparallel \varphi_3$. **Right)** The PDFA constructed by our algorithm for the LTL$_f$ formulas and the preference over them. The output of each state—the set of formulas that satisfies every string that ends at that state—is shown in blue and the most preferred formulas for each state is shown in red.

5 Synthesizing a Most-Preferred Policy

With the computational model PDFA representing the partially-ordered temporal goals, we are ready to present a planning algorithm to solve Problem 1. The algorithm computes for a given TLMDP, a policy that is most-preferred given the user preferences specified by a given PDFA. The first step is to augment the planning state space with the states 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, defined as follows.

Definition 14. Product MDP. *Let* $M = \langle S, A := \Sigma_{s \in S} A_s, \mathbf{P}, s_0, s_\perp, \mathcal{AP}, L \rangle$ *and* $\mathcal{A} = \langle Q, \Sigma, \delta, \iota, G := (\mathcal{W}, E) \rangle$ *be respectively the TLMDP and the PDFA. The product of* M *and* $\mathcal{A}$ *is a tuple* $\mathcal{M} = (X, A := \bigcup_{x \in X} A_x, \mathbf{T}, x_0, X_G, \mathcal{G} := (\mathcal{Y}, \mathcal{E}))$ *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. $\mathbf{T} : X \times A \to \mathcal{D}(X)$ *is the transition function such that for each state* $(s, q) \in X$, *action* $a \in A$, *and state* $(s', q') \in X$;

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

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. $\mathcal{G} = (\mathcal{Y}, \mathcal{E})$ *is the preference graph, in which, letting* $Y_i = \{s_\perp\} \times W_i$ *for each* $W_i \in \mathcal{W}$,
   - $\mathcal{Y} = \{Y_i \mid i = 1, \ldots, |\mathcal{W}|\}$ *is the vertex set of the graph, and*
   - $\mathcal{E}$ *is the edge set of the graph such that* $(Y_i, Y_j) \in \mathcal{E}$ *if and only if* $(W_i, W_j) \in E$.

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

Example 4. *Continuing with the example in Figure 2, we have* $Y_1 = \{s_\perp\} \times W_1 = \{(s_\perp, q_2)\}$, $Y_2 = \{s_\perp\} \times W_2 = \{(s_\perp, q_4)\}$, $Y_3 = \{s_\perp\} \times W_3 = \{(s_\perp, q_1)\}$, *and* $Y_4 = \{s_\perp\} \times W_4 = \{(s_\perp, q_0), (s_\perp, q_3), (s_\perp, q_5)\}$.

Next, we show how to compute a stochastic nondominated policy in the sense of Definition 8 through solving a Multi-objective MDP (MOMDP). The existence of such a MOMDP is guaranteed by the multi-utility representation theorem (Ok et al. 2002, Proposition 1), which states that for every partial order $\langle U, \succeq \rangle$ defined over a finite set U, there exists a vector-valued utility function $\mathbf{u} : U \to \mathbb{R}^n$ such that for any $x, y \in U$, $x \succeq y$ if and only if $\mathbf{u}(x) \geq \mathbf{u}(y)$ where $\geq$ is element-wise *.

We extend the notions related to stochastic ordering for state subsets of the product MDP $\mathcal{M}$, constructed in Definition 14, as follows:

Definition 15. *Let* $\mathbf{Y} \subseteq \mathcal{Y}$ *be a set of vertices in the preference graph* $\mathcal{G}$. *The upper closure of* $\mathbf{Y}$ *with respect to* $\mathcal{G}$ *is defined by*

$$\mathbf{Y}_i^{\uparrow} = \{Y' \mid \exists Y \in \mathbf{Y}, Y \rightsquigarrow Y'\}$$

and the lower closure of Y *is defined by*

$$\mathbf{Y}_i^{\downarrow} = \{Y' \mid \exists Y \in \mathbf{Y}, Y' \rightsquigarrow Y\}.$$

Also, $\mathbf{Y} \subseteq \mathcal{Y}$ *is called an* increasing set *if* $\mathbf{Y} = \mathbf{Y}^{\uparrow}$.

These sets are used to define a stochastic ordering type as follows:

Definition 16. *Let* $\mathfrak{E}_{st}(\mathcal{Y}), \mathfrak{E}_{wk}(\mathcal{Y}), \mathfrak{E}_{wk*}(\mathcal{Y})$ *denote the strong, weak, and weak* orderings, respectively, where*

$$\mathfrak{E}_{st}(\mathcal{Y}) = \{ \textit{all increasing sets in } 2^{\mathcal{Y}}\},$$
$$\mathfrak{E}_{wk}(\mathcal{Y}) = \{\{Y\}^{\uparrow} \mid Y \in \mathcal{Y}\} \cup \{\mathcal{Y}, \emptyset\},$$
$$\mathfrak{E}_{wk*}(\mathcal{Y}) = \{\mathcal{Y} \setminus \{Y\}^{\downarrow} \mid Y \in \mathcal{Y}\} \cup \{\mathcal{Y}, \emptyset\},$$

For a stochastic ordering $\mathfrak{E} \in \{\mathfrak{E}_{st}, \mathfrak{E}_{wk}, \mathfrak{E}_{wk*}\}$, let the elements of set $\mathfrak{E}(\mathcal{Y}) \setminus \{\mathcal{Y}, \emptyset\}$ to be indexed arbitrary as $\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_N$. We use this indexed set in the following construction to make a multi-objective MDP.

Definition 17. MOMDP. *Given a stochastic ordering* $\mathfrak{E} \in \{\mathfrak{E}_{st}, \mathfrak{E}_{wk}, \mathfrak{E}_{wk*}\}$, *the multi-objective MDP (MOMDP) associated with the product MDP* $\mathcal{M} = \langle X, A, \mathbf{T}, x_0, X_G, \mathcal{G} := (\mathcal{Y}, \mathcal{E}) \rangle$ *in Definition 14 and the stochastic ordering* $\mathfrak{E}$, *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* $\mathbf{Y}_i \in \mathfrak{E}(\mathcal{Y}) \setminus \{\mathcal{Y}, \emptyset\}$, $Z_i = \bigcup_{Y \in \mathbf{Y}_i} Y$. *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.

Remark 2. *We exclude* $\mathcal{Y}$ *and* $\emptyset$ *from the construction of the multi-objective MDP. This is because we consider only proper policies and under any proper policy, any state in* X_G *is reached with probability one. As a result, the probability of reaching the objectives* $\emptyset$ *and* $\mathcal{Y}$ *are always* 0 *an* 1, *respectively, regardless of the chosen stochastic ordering.*

To illustrate the construction of $\mathcal{P}$, we continue with our running example.

Example 5. *Using the running example in Figure 2, for which the preference graph of the product MDP is shown in Example 4, we have* $\{Y_1\}^{\uparrow} = \{Y_1\}$, $\{Y_2\}^{\uparrow} = \{Y_1, Y_2\}$, $\{Y_3\}^{\uparrow} = \{Y_1, Y_3\}$, *and* $\{Y_4\}^{\uparrow} = \{Y_1, Y_2, Y_3, Y_4\}$, *and as a result,*

$$\mathfrak{E}_{wk}(\mathcal{Y}) \setminus \{\mathcal{Y}, \emptyset\} = \{\{Y_1\}, \{Y_1, Y_2\}, \{Y_1, Y_3\}\},$$

and thus, under weak-stochastic ordering, the MOMDP will have the following objectives

$$Z_1 = Y_1, Z_2 = Y_1 \cup Y_2, \textit{and } Z_3 = Y_1 \cup Y_3.$$

Also, we have $\{Y_1\}^{\downarrow} = \{Y_1, Y_2, Y_3, Y_4\}$, $\{Y_2\}^{\downarrow} = \{Y_2, Y_4\}$, $\{Y_3\}^{\downarrow} = \{Y_3, Y_4\}$, *and*

$$\mathfrak{E}_{wk*}(\mathcal{Y}) \setminus \{\mathcal{Y}, \emptyset\} = \{\{Y_1, Y_3\}, \{Y_1, Y_2\}, \{Y_1, Y_2, Y_3\}\},$$

and therefore, under weak-stochastic ordering, the MOMDP will have the following objectives*

$$Z_1 = Y_1 \cup Y_3, Z_2 = Y_1 \cup Y_2, \textit{and } Z_3 = Y_1 \cup Y_2 \cup Y_3.$$

Furthermore,

$$\mathfrak{E}_{st}(\mathcal{Y}) \setminus \{\mathcal{Y}, \emptyset\} = \{\{Y_1\}, \{Y_1, Y_2\}, \{Y_1, Y_3\}, \{Y_1, Y_2, Y_3\}\},$$

and hence, under strong-stochastic ordering, the MOMDP will have the following objectives

$$Z_1 = Y_1, Z_2 = Y_1 \cup Y_2, Z_3 = Y_1 \cup Y_3, \textit{and } Z_3 = Y_1 \cup Y_2 \cup Y_3.$$

Accordingly, the objectives for each of the stochastic orderings in terms of the temporal goals in Figure 1 can be summarized as Table 1.

Table 1. Objectives of different stochastic ordering types for the temporal goals in Figure 1.

Stochastic Ordering	Objectives
Weak	$\{p_1\}, \{p_1, p_2\}, \{p_1, p_3\}$
Strong	$\{p_1\}, \{p_1, p_2\}, \{p_1, p_3\}, \{p_1, p_2, p_3\}$
Weak*	$\{p_1, p_2\}, \{p_1, p_3\}, \{p_1, p_2, p_3\}$

*Specifically, the multi-utility representation theorem (Ok et al. 2002, Proposition 1) requires the partial order $\succeq$ over the set U to be representable as an intersection of finitely many linear orders. However, Dushnik and Miller (1941); Fishburn (1985) have proved that every partial order over a finite set can be represented as an intersection of finitely many linear orders.

Given the MOMDP in Definition 17, 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 equation

$$\mathbf{V}^\mu(x) = \sum_{a \in A} \left(\mu(x, a) \sum_{x' \in X} \mathbf{T}(x, a, x') \mathbf{V}^\mu(x') \right). \quad (3)$$

Definition 18. *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_j without reducing the probability of reaching other sets Z_i's.

Definition 19. *A proper policy μ for the MOMDP is* Pareto optimal *if for no proper policy μ' for the MOMDP it holds that $\mu' > \mu$.*

In 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 (Chatterjee et al. 2006). Thus, we restrict to computing memoryless policies.

With this in mind, we present the following result.

Theorem 2. *Assume the MOMDP $\mathcal{P}$ in Definition 17 is constrcuted under a stochastic ordering $\mathfrak{E} \in \{\mathfrak{E}_{st}, \mathfrak{E}_{wk}, \mathfrak{E}_{wk*}\}$ and 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 $\pi(\rho) = \mu((s_n, \delta(\iota, \text{trace}(\rho))))$. If μ is Pareto optimal, then π is $\mathfrak{E}$-stochastic nondominated, respecting the preference specified by PDFA $\mathcal{A}$.*

Proof. We first provide a detailed proof of the case where $\mathfrak{E} = \mathfrak{E}_{wk}$, that is, where $\mathcal{P}$ is constructed for weak-stochastic ordering. We show that if μ is Pareto optimal then π is weak-stochastic nondominated. To facilitate the proof, the following notation is used: Let $\mathbf{Pr}^\mu(\text{reach}(H), \mathcal{M})$ be the probability of terminating in the set $H \subseteq X$ given the policy μ for the MOMDP and $\mathbf{Pr}^\pi(\text{reach}(P), M)$ be the probability of terminating in the set $P \subseteq S$ given the policy π in the original STLMDP.

First, consider that by the construction of the product MDP, Definition 14, preference graphs $\mathcal{G}$ and G are isomorphic, and thus, each $Y_i \in \mathcal{Y}$ is mapped to a single $W_i \in \mathcal{W}$, and vice versa. Define $W_i^+ = \bigcup_{W : W_i \rightsquigarrow W} W$ for each $W_i \in \mathcal{W}$. That is, let W_i^+ include the unions of states in all the nodes that can be reached from W_i in the preference graph. Note that $W_i \in W_i^+$. Given that $\mathcal{G}$ and G are isomorphic, $Y_i \rightsquigarrow Y_j$ if and only if $W_i \rightsquigarrow W_j$ for all $i, j \in \{1, 2, \cdots, N\}$. This combined with that $Z_i = \bigcup_{Y \in \{Y_i\}^\uparrow} Y$ for $i \in \{1, \cdots, N\}$ by Definition 17, implies that for each i,

$$\mathbf{V}_i^\mu = \mathbf{Pr}^\mu(\text{reach}(Z_i), \mathcal{M}) = \mathbf{Pr}^\pi(\text{reach}(W_i^+), M\}. \quad (4)$$

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 5, for each W_i and $w \in \Sigma^*$ such that $\delta(\iota, w) \in W_i$,

$$\mathbf{Pr}^\pi(\text{reach}(W_i^+), M) = \mathbf{Pr}^\pi(\{w\}^\uparrow). \quad (5)$$

Finally, given that μ is a Pareto optimal policy, by Definition 18 and Definition 19, 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 (4) and (5) 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 7 and Definition 8, means that π is weak-stochastic nondominated.

Proof for the case where $\mathcal{P}$ is constructed for weak*-stochastic ordering, that is, where $\mathfrak{E} = \mathfrak{E}_{wk*}$, is very similar, except that wherever W^+ is used we use $\overline{W}^-$, defined as $\overline{W_i}^- = \mathcal{W} \setminus \left(\bigcup_{W, W \rightsquigarrow W_i} W \right)$. For strong-stochastic ordering, we first construct for all subset $V \subseteq W$, the set $V^\uparrow = \{W \mid \exists W_i \in X, W_i \rightsquigarrow W\}$, then exclude any set $V^\uparrow$ if $V^\uparrow \neq V$. Then, for each remaining set $V^\uparrow$, the set of states contained in $V^\uparrow$ is used to define one reachability objective.

Given the MOMDP, one can use any existing methods to compute a set of Pareto optimal policies for $\mathcal{P}$. For a survey of those methods, see Roijers et al. (2013). 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.

6 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— corresponding to moving to the cell in the North, South, East, and West side of the current cell, respectively—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: Case 1 without stochasticity in the robot's dynamics, and Case 2 with stochasticity. In Case 1, when the robot decides to perform an action to move to a neighboring cell, its actuators will guarantee that the robot will move to that cell after performing the action. In Case 2, 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 i9, 2.50GHz CPU and a 32GB memory.

6.1 The Preference DFA

We first describe how the PDFA in Figure 2 is generated given the preference over LTL$_f$ formulas p_1-p_4 from Example 1, formulating the preferences in Figure 1. Figure 4 shows for each of the four LTL$_f$ formula, a DFA that encodes that formula. This figure also shows the PDFA our algorithm constructs for these four formulas and the associated user preferences, consisting of the underlying DFA and the preference graph, shown respectively in Part (e) and Part (f) of this figure. The PDFA is generated using our open-source tool [†] implemented in Python3. Note that the PDFA in this figure is the same PDFA in Figure 2, but this figure illustrates how the states q_0-q_5 in Figure 2 are constructed. The states in the underlying DFA represent the formulas satisfied by any word whose trace ends in that state. For instance, $\mathsf{Outcomes}((3,2,3,4)) = \{p_4\}$. This is because the first three components $3, 2, 3$ are not accepting in their respective DFAs, but the fourth component, 4, is an accepting state in the DFA for p_4 (see Figure 4). The states of the preference graph encode the partition of Q based on the most-preferred outcomes satisfied by the states. For example, state $(3, 6, 3, 2)$ belongs to the block $(0, 1, 0, 0)$ since the most-preferred outcomes satisfied by $(3, 6, 3, 2)$ is $\{p_2\}$.

Recall how the PDFA produces a ranking over the words in Σ^*. For example, consider two paths in the MDP M: A path ρ_1 that pollinates tulips first and then daisies, and a path ρ_2 that first pollinates orchids and then daisies. The trace of ρ_1, $\mathrm{trace}(\rho_1)$, induces a path that terminates in state $(4, 5, 3, 2)$ of the underlying DFA, whereas $\mathrm{trace}(\rho_2)$ terminates in $(3, 6, 3, 2)$. Accordingly, state $(4, 5, 3, 2)$ of the PDFA belongs to the block of the partition represented by the vertex $(1, 0, 0, 0)$ in the preference graph, and state $(3, 6, 3, 2)$ belongs to the block represented by the vertex $(0, 1, 0, 0)$. Since the preference graph has an edge from $(0, 1, 0, 0)$ to $(1, 0, 0, 0)$, it is implied that ρ_1 is strictly preferred over ρ_2.

6.2 Case 1: Deterministic Action Transitions but Uncertain Environments

For the case when the robot's actions have deterministic outcomes, the constructed MDP has $10,460$ states and $280,643$ transitions (its transition function has $280,643$ entries with non-zero probabilities). It took 39.06 seconds for our program to construct the MDP. The product MDP had $36,649$ states and $946,467$ transitions. The average construction times for the product MDP over 10 constructions for each of the weak stochastic order, strong stochastic order, and weak* stochastic order were respectively 238.78, 238.57, and 238.76 seconds.

Given the preference described in Fig. 2, we employ linear scalarization methods to solve the MOMDP. Specifically, given a weight vector $\mathbf{w} \in [0,1]^N$, we compute the nondominated policy $\mu_{\mathbf{w}}$, by first setting $V_{\mathbf{w}}(x) = \sum_{1 \le i \le N : x \in Z_i} \mathbf{w}[i]$ for each goal state $x \in X_G$, and then by solving the following Bellman equation for the values of the non-goal states $x \in X \setminus X_G$:

$$V_{\mathbf{w}}(x) = \max_{a \in A_x} \sum_{x' \in X} \mathbf{T}(x, a, x') V_{\mathbf{w}}(x'). \tag{6}$$

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

$$\mu_{\mathbf{w}}(x) = \operatorname*{argmax}_{a \in A_x} \sum_{x' \in X} \mathbf{T}(x, a, x') V_{\mathbf{w}}(x'). \tag{7}$$

For each of the three stochastic orderings $\mathfrak{E} \in \{\mathfrak{E}_{st}, \mathfrak{E}_{wk}, \mathfrak{E}_{wk*}\}$, we randomly generated $1,000$ weight vectors and used each one of them to compute a Pareto optimal policy for the MOMDP. For each stochastic orderings $\mathfrak{E}$, the $1,000$ computed Pareto-optimal policies are expected to be $\mathfrak{E}$-nondominated policies. From the result, it is noted that for each stochastic orderings $\mathfrak{E}$, none of those $1,000$ computed polices were $\mathfrak{E}$-stochastic dominated by the other polices. This is expected due to Theorem 2.

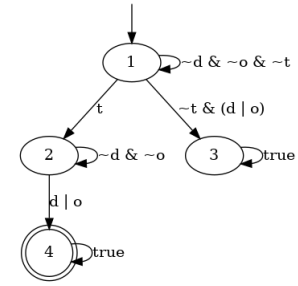
Next, we provide more detailed analysis for weak-stochastic non-dominated policies. Recall from Table 1 that the objectives for the weak-stochastic ordering are $\{p_1\}$, $\{p_1, p_2\}$, and $\{p_1, p_3\}$. Since it is difficult to illustrate the 3D Pareto front, we select a set of policies with similar probabilities (approximately 0.24) of satisfying p_1 and then plot the values of these policies for the objectives $\{p_1, p_2\}$ and $\{p_1, p_3\}$. Figure 5 shows the values of those policies for the objectives $\{p_1, p_2\}$ and $\{p_1, p_3\}$. This figure shows that none of those policies weak-stochastic dominates each other.

	Weight Vector	Value Vector $[\{p_1\}, \{p_1, p_2\}, \{p_1, p_3\}]$	Prob. of individual outcomes
1	[0.466, 0.412, 0.122]	[0.110, 0.799, 0.291]	[0.110, 0.689, 0.181, 0.020]
2	[0.363, 0.438, 0.199]	[0.146, 0.726, 0.395]	[0.146, 0.580, 0.249, 0.025]
3	[0.207, 0.484, 0.309]	[0.201, 0.558, 0.633]	[0.201, 0.357, 0.432, 0.01]
4	[0.134, 0.519, 0.347]	[0.173, 0.638, 0.527]	[0.173, 0.465, 0.354, 0.008]
5	[0.141, 0.541, 0.318]	[0.068, 0.874, 0.187]	[0.068, 0.806, 0.119, 0.007]
6	[0.434, 0.339, 0.227]	[0.241, 0.427, 0.798]	[0.241, 0.186, 0.557, 0.016]
7	[0.428, 0.223, 0.349]	[0.239, 0.250, 0.980]	[0.239, 0.011, 0.741, 0.009]
8	[0.213, 0.395, 0.392]	[0.240, 0.307, 0.925]	[0.240, 0.067, 0.685, 0.008]
9	[0.742, 0.208, 0.050]	[0.240, 0.432, 0.787]	[0.240, 0.192, 0.547, 0.021]
10	[0.057, 0.488, 0.455]	[0.240, 0.398, 0.831]	[0.240, 0.158, 0.591, 0.011]

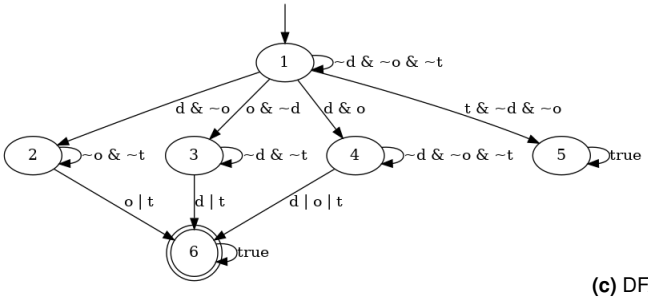
Table 2. Ten weak-stochastic nondominated polices computed by our algorithm for the Garden case study.

Table 2 shows 10 out of those $1,000$ weight vectors along with the value vectors of the weak-stochastic non-dominated polices computed for those weight vectors and the corresponding probabilities those polices assign to the four

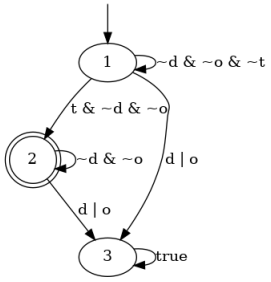
[†] Tool for constructing PDFA from a preference over LTL$_f$ formulas: https://akulkarni.me/prefltlf2pdfa.html



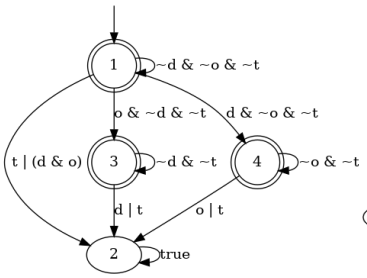
(a) DFA for p_1: $(\neg d \wedge \neg o)\,\mathsf{U}\,(t \wedge \bigcirc \Diamond (d \vee o))$.



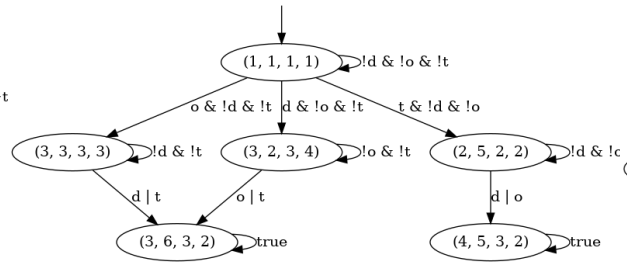
(b) DFA for p_2: $\neg t\,\mathsf{U}\,((o \wedge \bigcirc \Diamond (d \vee t)) \vee (d \wedge \bigcirc \Diamond (o \vee t)))$.



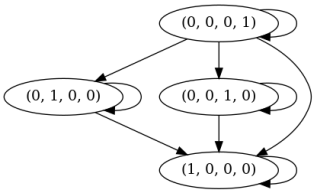
(c) DFA for p_3: $(\neg d \wedge \neg o)\,\mathsf{U}\,(t \wedge \Box (\neg d \wedge \neg o))$.



(d) DFA for p_4: $\Box (\neg d \wedge \neg o \wedge \neg t) \vee (\Diamond o \wedge \Box (\neg d \wedge \neg t)) \vee (\Diamond d \wedge \Box (\neg o \wedge \neg t))$.



(e) The underlying DFA of the PDFA.



(f) The preference graph of the PDFA.

Figure 4. a)-d) The DFAs for $p_1 - p_4$ for the example Figure 1, which are constrcuted by our online tools, available at https://akulkarni.me/prefltlf2pdfa.html. **e)-f)** The PDFA for the example in Figure 1, which is constructed the implementation of our algorithm for converting a preference model over LTL$_f$ formulas into a PDFA.

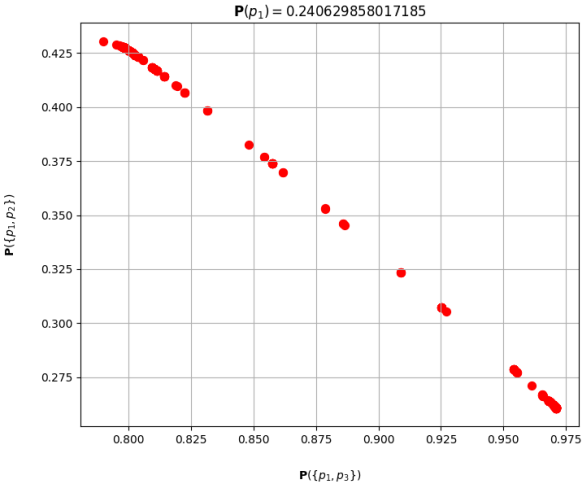


Figure 5. The probabilities of satisfying objectives $\{p_1, p_2\}$ and $\{p_1, p_3\}$ by the computed policies for weak-stochastic ordering who satisfy the objective $\{p_1\}$ with probability 0.2406.

preferences p_1 through p_4. For each policy, the last column shows a probability distribution over individual outcomes $p_1, \ldots, p_4$ indicating the probabilities of satisfying those formulas (in that 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 6 and 9 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 and not encounter rain 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 very small, regardless of the weight vector. 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 eighth and the tenth 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, even 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.

6.3 Case 2: Introducing Stochastic Robot Dynamics

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. We consider all the three types of stochastic orderings. The MDP construction time for weak-stochastic ordering, strong-stochastic ordering, and weak*-stochastic ordering were respectively 205.92, $203,43$ and 238.83 seconds. The construction time of the product MDP for these three stochastic ordering types were respectively $1,088.58$, $1,088.59$, and $1,089.02$ seconds.

Due to the stochasticity in the robot's dynamic, for each kind of stochastic ordering, we expect the policy computed for a specific weight vector to perform "poorer" than a policy computed for the same weight vector of Case 1. We first compare the two polices for weak-stochastic ordering, given the weight vector $[0.3333, 0.3333, 0.3334]$. The probabilities of the preferences to be satisfied for the variant without stochasticity were $[p_1 : 0.241, p_2 : 0.053, p_3 : 0.699, p_4 : 0.007]$, while those probabilities for the variant with stochasticity were $[p_1 : 0.008, p_2 : 0.821, p_3 : 0.120, p_4 : 0.051]$. While the former policy yields a higher probability of achieving p_3, the latter policy puts most of its efforts to satisfy p_2 and has a very low probability (close to 0) to satisfy the most preferred goal p_1. Similar observation is made for strong-stochastic ordering using 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 $[p_1 : 0.241, p_2 : 0.053, p_3 : 0.699, p_4 : 0.007]$, while those probabilities for the variant with stochasticity were $[p_1 : 0.002, p_2 : 0.931, p_3 : 0.026, p_4 : 0.041]$.

Lastly, for weak*-stochastic ordering, we have three objectives $\{p_1, p_2\}$, $\{p_1, p_3\}$, $\{p_1, p_2, p_3\}$ (after removing the empty set and the set $\{p_1, p_2, p_3, p_4\}$). Given the weight vector $[0.3333, 0.3333, 0.3334]$, the probabilities of the preferences to be satisfied for the variant without stochasticity were $[p_1 : 0.241, p_2 : 0.053, p_3 : 0.699, p_4 : 0.007]$, while those probabilities for the variant with stochasticity were $[p_1 : 0.000, p_2 : 0.953, p_3 : 0.007, p_4 : 0.040]$.

Given the same weight vector but different stochastic orderings, we observed that the probability of satisfying p_2 in the stochastic variant is much larger (≈ 16 times more likely) than that of the deterministic variant. This result is mainly due to the difficulty in reaching tulips given the coupled inherent stochastic dynamics and uncertain environmental factors (clouds and the bird). Because the chance of reaching tulips is very small, the probability of satisfying p_1 or p_3 – both require tulips to be visited — are equally small. As a result, the preference-based planner (across all three stochastic orders) satisfies p_2 with a much higher probability since p_2 only requires two flower types to be pollinated. This experimental comparison demonstrates the flexibility of preference-based planners to adjust the goal based on changes in the system and environment dynamics.

7 Conclusions and Future Work

In this paper, we introduced a formal language for specifying user's partially-ordered preferences over temporal goals expressed in LTL_f. We developed an algorithm to convert the user preference over LTL_f formulas into an automaton with a preorder over the acceptance conditions. To synthesize a most preferred policy in a stochastic environment, we utilized stochastic ordering to translate a partially-ordered user's preference to a preorder over probabilistic distributions over the system trajectories. This allowed us to rank the policies based on the partially-ordered user's preference. Leveraging the automaton structure, we proved that computing a most-preferred policy can be reduced to finding a Pareto-optimal policy in a multi-objective MDP augmented with the automaton states.

This work provides fundamental algorithms and principled approach for preference-based probabilistic planning with partially-ordered temporal logic objectives in stochastic systems. A direction for future work will be to extend 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 robotic applications, it would be of practical interest to design a conversational-AI interface that elicits human preferences and translating natural language specifications into the preference model, and thus facilitate human-on-the-loop planning.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Air Force Office of Scientific Research under award number FA9550-21-1-0085 and in part by NSF under award numbers 2024802 and 2144113.

References

Amorese P and Lahijanian M (2023) Optimal cost-preference trade-off planning with multiple temporal tasks. *arXiv preprint arXiv:2306.13222* .

Aumann RJ (1962) Utility theory without the completeness axiom. *Econometrica: Journal of the Econometric Society* : 445–462.

Baier C and Katoen JP (2008) *Principles of model checking*. MIT press.

Baier JA and McIlraith SA (2008) Planning with Preferences. *AI Magazine* 29(4): 25. DOI:10.1609/aimag.v29i4.2204.

Bertsekas DP and Tsitsiklis JN (1991) An analysis of stochastic shortest path problems. *Mathematics of Operations Research* 16(3): 580–595.

Bienvenu M, Fritz C and McIlraith SA (2011) Specifying and computing preferred plans. *Artificial Intelligence* 175(7-8): 1308–1345.

Cai M, Xiao S, Li Z and Kan Z (2021) Optimal probabilistic motion planning with potential infeasible LTL constraints. *IEEE Transactions on Automatic Control* 68(1): 301–316.

Cardona GA, Kamale D and Vasile CI (2023) Mixed integer linear programming approach for control synthesis with weighted signal temporal logic. In: *Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control*. pp. 1–12.

Chatterjee K, Majumdar R and Henzinger TA (2006) Markov decision processes with multiple objectives. In: *Annual*

symposium on theoretical aspects of computer science. Springer, pp. 325–336.

Chen Y, Gandhi R, Zhang Y and Fan C (2023) Nl2tl: Transforming natural languages to temporal logics using large language models. *arXiv preprint arXiv:2305.07766* .

Cormen TH, Leiserson CE, Rivest RL and Stein C (2022) *Introduction to algorithms*. MIT press.

Cosler M, Hahn C, Mendoza D, Schmitt F and Trippel C (2023) nl2spec: Interactively Translating Unstructured Natural Language to Temporal Logics with Large Language Models. In: Enea C and Lal A (eds.) *Computer Aided Verification*, Lecture Notes in Computer Science. Cham: Springer Nature Switzerland. ISBN 978-3-031-37703-7, pp. 383–396. DOI: 10.1007/978-3-031-37703-7_18.

De Giacomo G and Vardi MY (2013) Linear temporal logic and linear dynamic logic on finite traces. In: *IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. Association for Computing Machinery, pp. 854–860.

Dushnik B and Miller EW (1941) Partially ordered sets. *American journal of mathematics* 63(3): 600–610.

Fishburn PC (1985) Interval graphs and interval orders. *Discrete mathematics* 55(2): 135–149.

Fu J (2021) Probabilistic planning with preferences over temporal goals. In: *2021 American Control Conference (ACC)*. IEEE, pp. 4854–4859.

Hansson SO (2001) *The structure of values and norms*. Cambridge University Press.

Hansson SO and Grüne-Yanoff T (2022) *Preferences*. The Stanford Encyclopedia of Philosophy.

Hastie R and Dawes RM (2010) *Rational choice in an uncertain world: The psychology of judgment and decision making*. Sage.

Kulkarni AN and Fu J (2022) Opportunistic qualitative planning in stochastic systems with preferences over temporal logic objectives. *arXiv preprint arXiv:2203.13803* .

Lahijanian M and Kwiatkowska M (2016) Specification revision for Markov decision processes with optimal trade-off. In: *Proc. 55th Conference on Decision and Control (CDC'16)*. pp. 7411–7418.

Li L, Rahmani H and Fu J (2023) Probabilistic Planning with Prioritized Preferences over Temporal Logic Objectives. pp. 189–198. ISSN: 1045-0823.

Li M, Turrini A, Hahn EM, She Z and Zhang L (2020) Probabilistic preference planning problem for markov decision processes. *IEEE transactions on software engineering* .

Manna Z and Pnueli A (2012) *The temporal logic of reactive and concurrent systems: Specification*. Springer Science & Business Media.

Massey WA (1987) Stochastic Orderings for Markov Processes on Partially Ordered Spaces. *Mathematics of Operations Research* 12(2): 350–367. Publisher: INFORMS.

Mehdipour N, Vasile CI and Belta C (2021) Specifying User Preferences Using Weighted Signal Temporal Logic. *IEEE Control Systems Letters* 5(6): 2006–2011. DOI:10.1109/LCSYS.2020.3047362.

Ok EA et al. (2002) Utility representation of an incomplete preference relation. *Journal of Economic Theory* 104(2): 429–449.

Rahmani H, Kulkarni AN and Fu J (2023) Probabilistic planning with partially ordered preferences over temporal goals. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5702–5708.

Rahmani H and O'Kane JM (2019) Optimal temporal logic planning with cascading soft constraints. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2524–2531.

Rahmani H and O'Kane JM (2020) 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)*. IEEE, pp. 6619–6626.

Roijers DM, Vamplew P, Whiteson S and Dazeley R (2013) A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research* 48: 67–113.

Santhanam GR, Basu S and Honavar V (2016) Representing and Reasoning with Qualitative Preferences: Tools and Applications. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 10(1): 1–154.

Tumova J, Hall GC, Karaman S, Frazzoli E and Rus D (2013) Least-violating control strategy synthesis with safety rules. In: *Proceedings of the 16th international conference on Hybrid systems: computation and control*. ACM, pp. 1–10.

Wongpiromsarn T, Slutsky K, Frazzoli E and Topcu U (2021) Minimum-violation planning for autonomous systems: Theoretical and practical considerations. In: *2021 American Control Conference (ACC)*. pp. 4866–4872. DOI:10.23919/ACC50511.2021.9483174.