Preferences on Partial Satisfaction using Weighted Signal Temporal Logic Specifications

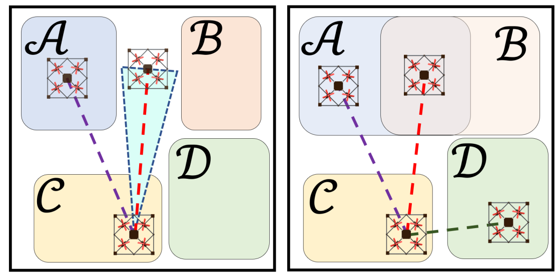
Gustavo A. Cardona and Cristian-Ioan Vasile

Abstract— **This work presents partial satisfaction control synthesis over an extension of Weighted Signal Temporal Logic wSTL called wSTL+. The new specification language wSTL+ enables the definition of preferences and importance of subformulae as weights over-inclusive (soft) operators (i.e., standard Boolean and temporal operators from wSTL). Furthermore, it includes exclusive operators that impose hard constraints to disallow specific subformulas to be partially satisfied. All subformulae must be fully satisfied or violated for conjunctive operators (conjunction and always). In the case of disjunctive operators (disjunction and eventually), mutual exclusive satisfaction is imposed, i.e., exactly one subformula holds. The weights in the specification capture the preferences and importance of fully satisfiable specifications and modulate the solution over conflicting or infeasible specifications. We formulate the partial satisfaction problem over wSTL+ specifications as a bilevel optimization problem. The inner level is modeled as a MILP and captures the customized satisfaction of the wSTL+ specification. The outer level is a linear program that maximizes the robustness of the satisfiable solution found in the inner level. Finally, we show the performance of our method in different case studies involving robot navigation in planar environments.**

I. Introduction

In recent years, Temporal Logic formalisms have proved to be useful for efficiently handling and scheduling complex tasks [Bellini et al.(2000)Bellini, Mattolini, and Nesi], [Kress-Gazit et al.(2018)Kress-Gazit, Lahijanian, and Raman] due to their rich expressivity to specify complex temporal and logical system behavior. Linear Temporal Logic (LTL) allows imposing Boolean and temporal constraints [Baier and Katoen(2008)], [Bisoffi and Dimarogonas(2020)]. However, the most significant limitation comes from considering implicit time. Therefore, timed specification languages, such as Signal Temporal Logic (STL), have been proposed to capture time explicitly [Maler and Nickovic(2004)], [Sun et al.(2022)Sun, Chen, Mitra, and Fan]. Additionally, STL is defined over continuous signals and real predicates and provides quantitative semantics (robustness) indicating the margin of satisfaction or violation of the specification [Haghighi et al.(2019)Haghighi, Mehdipour, Bartocci, and Belta], [Lindemann and Dimarogonas(2019b)], [Mehdipour et al.(2020)Mehdipour, Vasile, and Belta]. Hence, the control synthesis using STL can be modeled as an optimization problem where maximizing robustness leads to optimal satisfaction [Sadraddini and Belta(2015)], [Leahy et al.(2021)Leahy, Serlin, Vasile, Schoer, Jones, Tron, and Belta].

Gustavo A. Cardona and Cristian-Ioan Vasile are with the Mechanical Engineering and Mechanics Department at Lehigh University, PA, USA:{gcardona, cvasile}@lehigh.edu



(a) (b)

Fig. 1. Drone tasked to visit multiple regions in a planar environment. (a) comparison of solutions for infeasible specification in STL, wSTL, and wSTL+. (b) exclusive operators' versatility in wSTL+ specification.

Although STL is a very versatile language, every subformula in the specification has the same importance. Nevertheless, in reality, there are situations where one formula or time intervals are preferable over others. Therefore, an extension to STL was proposed in [Mehdipour et al.(2020)Mehdipour, Vasile, and Belta] called Weighted Signal Temporal Logic (wSTL) that captures preferences, priorities, and importance of subformulae by incorporating weights over logical and temporal operators. In this work, we proposed an extension to wSTL, referred to as wSTL+, where exclusive operations are added to the Boolean and Temporal operators. These capture that all subformulae are counted into the satisfaction and at all times or not considered at all in case of *exclusive conjunction* and *exclusive always*, respectively. In the case of *exclusive disjunction* (*exclusive eventually*), one subformula is satisfied (at a one-time step only), and no other is satisfied (at no other time step within the interval).

On the other hand, even though STL and wSTL standard robustness definitions can compute a minimally violated solution in case of conflicting or infeasible specifications, none of them can partially satisfy the conflicting formulae in the specification. In this work, we propose a framework that captures wSTL+ semantics in the context of partial satisfaction. To make this clear, let us consider the following example:

Example 1. *A quadrotor in a planar environment, shown in Fig. 1(a) with disjoint labeled regions of interest, is tasked with the following mission specification: "Always between 2 to 4 hours after deployment monitor region $\mathcal{A}$ and $\mathcal{B}$."*

In Fig. 1(a) shows the expected computed solution by STL (red dashed line) and wSTL (cyan-filled area) corresponding to the minimal violation of the specification (staying in the middle of $\mathcal{A}$ and $\mathcal{B}$) in the case of STL and the span of a deviated violation solution in the case of the wSTL depending on the assigned preference weight, respectively. Note that none of them actually satisfies in full or partially the specification. The desired solution is the partial satisfaction of the specification given as wSTL+ (purple dashed line). Furthermore, thanks to the inclusion of exclusive operators in wSTL+, it can impose the partial satisfaction solution even if the regions of interest are no longer disjoint sets that must not be visited simultaneously. In Fig. 1(b) shows solutions corresponding to *exclusive disjunction* (purple), both *exclusive conjunction* (red), and none *exclusive conjunction*.

Multiple methods in the literature tackle partial satisfaction in the context of temporal logic going from automata based approaches [Cai et al.(2020b)Cai, Peng, Li, and Kan], [Cai et al.(2020a)Cai, Peng, Li, Gao, and Kan], [Kamale et al.(2021)Kamale, Karyofylli, and Vasile], [Lacerda et al.(2015)Lacerda, Parker, and Hawes], [Lahijanian et al.(2015)Lahijanian, Almagor, Fried, Kavraki, and Vardi], [Guo and Dimarogonas(2015)], control barrier functions [Lindemann and Dimarogonas(2019a)] to computing policies in an optimization model [Raman et al.(2014)Raman, Donzé, Maasoumy, Murray, Sangiovanni-Vincentelli, and Seshia], [Choudhury et al.(2020)Choudhury, Gupta, Kochenderfer, Sadigh, and Bohg]. However, our primary focus is on the extension of our previous work [Cardona and Vasile(2023)] that captures the global partial satisfaction of STL specifications as a bilevel optimization problem. The inner level identifies satisfiable subformulae prioritizing lower-depth on the Abstract Syntax Tree (AST) modeled as a Mixed Integer Linear Program (MILP). The outer level is a Linear Program (LP) that maximizes the robustness of all satisfied subformulae by a solution of the MILP. In this work, we consider the partial satisfaction over wSTL+ specifications that allows us to modulate and capture preferences over the partial satisfiable solution. Differently from [Cardona and Vasile(2023)] that computed fractions of satisfaction, the solution of our inner level measures how close the solution is to the preferred specified solution.

The main contributions of this work are

1) We propose an extension to wSTL [Mehdipour et al.(2020)Mehdipour, Vasile, and Belta], referred to as wSTL+, including exclusive operators such as *exclusive conjunction*, *exclusive disjunction*, *exclusive always*, and *exclusive eventually*.
2) We propose a definition of the weights, Boolean, and temporal operators in wSTL+ in the context of partial satisfaction as tie-breaking rules for conflicting subformulae, inclusive (soft), and exclusive (hard) preference constraints.
3) We formulate a MILP approach that captures the semantics of wSTL+ specifications as fractions of preferred satisfaction.
4) Finally, we show the versatility of the wSTL+ to modulate the solution on conflicting or infeasible specifications and time performance in three case studies involving agents navigating in planar environments.

II. Preliminaries and Notation

Let $\mathbb{Z}$, $\mathbb{R}$, and $\mathbb{B}$ denote the sets of integer, real, and binary numbers. The set of integers greater than a is $\mathbb{Z}_{\geq a}$. For a set S, 2^S and $|S|$ represent its power set and cardinality. For $S \subseteq \mathbb{R}$ and $\alpha \in \mathbb{R}$, we have $\alpha + S = \{\alpha + x \mid x \in S\}$. The integer interval (range) from a to b is $[a..b]$. For a range $I = [a..b]$, we use $\underline{I} = a$ and $\bar{I} = b$. Let $x \in \mathbb{R}^d$ be a d-dimensional vector. The i-th component of x is given by x_i, $i \in [1..d]$.

A. Weighted Signal Temporal Logic

Here we describe the semantics of Weighted Signal Temporal Logic (wSTL) introduced in [Mehdipour et al.(2020)Mehdipour, Vasile, and Belta]. It is an extension of Signal Temporal Logic that allows specifications to capture user preferences, priorities, and importance associated with the Boolean and temporal operators.

Definition 1 (Weighted Signal Temporal Logic (wSTL) [Mehdipour et al.(2020)Mehdipour, Vasile, and Belta])**.** *The syntax of wSTL in [Mehdipour et al.(2020)Mehdipour, Vasile, and Belta] is defined in Backus-Naur form as follows*

$$\varphi ::= \top \mid \bot \mid \mu \mid \neg\varphi \mid \bigwedge_{i\in[1..N]}{}^{p}\varphi_i \mid \bigvee_{i\in[1..N]}{}^{p}\varphi_i \mid \Diamond_I^w \varphi \mid \Box_I^w \varphi, \tag{1}$$

where $\top$ and $\bot$ are the logical *True* and *False*; μ is a linear *predicate* of the form $s_i \geq \pi$ with threshold π over the i-th component of signal s; $\neg$, $\wedge$, and $\vee$ are the Boolean *negation*, *conjunction* and *disjunction* operators, respectively. $\Diamond$ (*eventually*) and $\Box$ (*always*) are temporal operators with time bound in the range I with the same definitions as for STL [Maler and Nickovic(2004)]. Weight functions assign the positive weights over Boolean operators' conjunction and disjunction formulae $p : [1..N] \to \mathbb{R}_{>0}$, where N is the number of sub-formulae under the operator. For sub-formulae in conjunction, denoted as $(\wedge^p(\varphi_1, \varphi_2, \dots, \varphi_N))$, the weights capture the importance of parallel tasks, whereas for the sub-formulae in disjunction, denoted as $(\vee^p(\varphi_1, \varphi_2, \dots, \varphi_N))$, the weights indicate priorities for alternatives. The positive weight functions $w : I \to \mathbb{R}_{>0}$ capture user preferences for satisfaction times for the eventually operator and the importance of satisfaction times in the case of the always operator.

wSTL specifications with all weights p and w equal one are equivalent to STL specifications. The *Boolean* (qualitative) semantics of a wSTL specification φ is the same as the STL specification ϕ, the unweighted version of wSTL.

The *robustness* (quantitative semantics) of a wSTL specification φ captures the margin of satisfaction or violation of a signal s over φ modulated by the specified weights.

Definition 2 (Weighted Traditional Robustness)**.** *Given a wSTL specification φ and a signal s, the weighted robustness* *score $\tilde{\rho}(\varphi, s, k)$ at time k is recursively defined as follows*

$$
\begin{aligned}
\tilde{\rho}(\mu, s, k) &:= s_i(k) - \pi, \\
\tilde{\rho}\,(\neg\varphi, s, k) &:= -\tilde{\rho}(\varphi, s, k), \\
\tilde{\rho}\left(\bigwedge_i{}^{p}\varphi_i, s, k\right) &:= \min_i \left\{p_i^{\wedge} \cdot \tilde{\rho}(\varphi_i, s, k)\right\}, \\
\tilde{\rho}\left(\bigvee_i{}^{p}\varphi_i, s, k\right) &:= \max_i \left\{p_i^{\vee} \cdot \tilde{\rho}(\varphi_i, s, k)\right\}, \\
\tilde{\rho}\,(\Box_I^w \varphi, s, k) &:= \min_{k' \in k+I} \left\{w^{\Box}(k'-k) \cdot \tilde{\rho}(\varphi, s, k')\right\}, \\
\tilde{\rho}\,(\Diamond_I^w \varphi, s, k) &:= \max_{k' \in k+I} \left\{w^{\Diamond}(k'-k) \cdot \tilde{\rho}(\varphi, s, k')\right\},
\end{aligned}
\tag{2}
$$

where $I' = [k..k']$, and $p_i^{\wedge}$, $p_i^{\vee}$, $w^{\Box}(k'-k)$, and $w^{\Diamond}(k'-k)$ are appropriate normalized weights.

Note that if all weights in the wSTL specification are positive, then the robustness score is a scaled value of the equivalent STL robustness.

Theorem 1 (wSTL Soundness [Mehdipour et al.(2020)Mehdipour, Vasile, and Belta])**.** *The robustness score of wSTL $\tilde{\rho}(\varphi, s, k)$ is sound iff*

$$
\begin{aligned}
\tilde{\rho}(\varphi, s, k) > 0 &\iff \rho(\phi, s, k) > 0 \to s \vDash \varphi, \\
\tilde{\rho}(\varphi, s, k) < 0 &\iff \rho(\phi, s, k) < 0 \to s \nvDash \varphi,
\end{aligned}
$$

Note that a positive robustness score of a wSTL specification φ implies that the robustness score of its equivalent STL specification ϕ is also positive and therefore signal s satisfies the specification.

The time horizon of an wSTL formula [Dokhanchi et al.(2014)Dokhanchi, Hoxha, and Fainekos] is defined as

$$
\|\varphi\| = \begin{cases}
0, & \text{if } \varphi = s_i \geq \pi, \\
\|\varphi_1\|, & \text{if } \varphi = \neg\varphi_1, \\
\max\{\|\varphi_1\|, \|\varphi_2\|\}, & \text{if } \varphi \in \{\wedge^p(\varphi_1, \varphi_2), \vee^p(\varphi_1, \varphi_2)\}, \\
\bar{I} + \|\varphi_1\|, & \text{if } \varphi \in \{\Diamond_I^w \varphi_1, \Box_I^w \varphi_1\}.
\end{cases}
\tag{3}
$$

A wSTL formula is said to be in *positive normal form* (PNF) if it satisfies two conditions. First, all its predicates are of the $s_i \geq \mu$ form. Second, it does not contain the negation operator. Any wSTL formula can be represented using an abstract syntax tree (AST) in which intermediate nodes correspond to logical and temporal operators, and leaves to predicates [Hopcroft et al.(2001)Hopcroft, Motwani, and Ullman], weights p and w are weights assigned to the edges of the tree. We use $\varphi' \sqsubset \varphi$ to denote that φ' is a proper subformula of φ, and $\varphi' \sqsubseteq \varphi$ when they can also be equal.

III. Problem Formulation

This section introduces the control synthesis problem subject to a mission specification imposing temporal and logical constraints. We focus on cases where not all parts of the mission can be satisfied due to control input limitations or conflicting (competing) specifications, and then a customized partial satisfaction of the mission is desired. We introduce a new language specification called wSTL+, an extension of wSTL that captures hard and soft constraints in the form of exclusive satisfaction and preferences and importance over partial satisfaction of specifications, respectively. In the case of fully satisfiable specifications, wSTL+ modulates the robustness degree based on specification weights. When only partial satisfaction can be achieved, wSTL+ provides finer specification control over which parts of the specification should be prioritized and how to break ties. We define two different types of preferences. First, *inclusive preferences* (all operators in standard wSTL) that indicate the preferable subformulae to be accounted into satisfaction if possible. Second, *exclusive preferences* indicates that the subformula needs to be entirely satisfied or not considered at all in the case of "conjunction" and "always" operators. For "disjunction" and "eventually" operators, just one and not any other subformulae (mutually exclusive).

Finally, we define the *partial satisfaction* (PS) problem that requires synthesizing control inputs such that as much of the formula is satisfied while capturing user-inclusive and exclusive preferences.

A. Weighted Signal Temporal Logic (wSTL+)

In this section, we introduce an extension of wSTL referred to as Weighted Signal Temporal Logic (wSTL+) that allows specifications to capture preferences, priorities, and importance inclusively and exclusively through the Boolean and temporal operators.

Definition 3 (Weighted Signal Temporal Logic (wSTL+))**.** *The syntax of wSTL+ in Backus-Naur form over linear predicates is*

$$
\begin{aligned}
\varphi ::= \top \mid \mu \mid \neg\varphi \mid \bigwedge_{i \in [1..N]}{}^{p}\varphi_i \mid \dot{\bigwedge_{i \in [1..N]}}{}^{p}\varphi_i \mid \bigvee_{i \in [1..N]}{}^{p}\varphi_i \mid \\
\dot{\bigvee_{i \in [1..N]}}{}^{p}\varphi_i \mid \Diamond_I^w \varphi \mid \dot{\Diamond}_I^w \varphi \mid \Box_I^w \varphi \mid \boxdot_I^w \varphi,
\end{aligned}
\tag{4}
$$

where the weights p, w, predicates μ, Boolean true $\top$, inclusive conjunction $\wedge_i^p$, and inclusive disjunction $\vee_i^p$, inclusive eventually $\Diamond_I^w$, and inclusive always $\Box_I^w$ are semantically identical to wSTL semantics (1). Note that we call these inclusive preferences since the weights p_i and w_i capture the preferences over the subformulae of the mission specification. Higher weight values of p_i will imply a higher preference over the i-th subformula, while w_i describes importance at time steps within interval I.

Example 2 (Continuation of Example (1))**.** *Thus, example (1) can now be expressed as $\varphi_1 = \Box_{[2,4]}^w(\wedge^p(\mathcal{A}, \mathcal{B}))$ as the desired solution in Fig. 1(a) (purple) can be accomplished by defining appropriate weights. For instance, $w = [1, 1, 1]$ since no specific timestep has a higher importance, and $p = [p_1, p_2]$ with $p_1 > p_2$ which will imply the preference for going to region $\mathcal{A}$.*

On the other hand, let us consider a signal s at time k as $s(k)$ and, for simplification, p and w equal to one for all operators. Then $\dot{\wedge}_i^p\varphi_i$ with $i \in [1..N]$ is the *exclusive conjunction*, meaning that a signal $s(k)$ has to satisfy every subformula φ_i or not at all. Formally, $s(k) \vDash \varphi \equiv (s(k) \vDash \varphi_i, \forall i \in [1..N]) \vee (s(k) \nvDash \varphi_i, \forall i \in [1..N])$. $\dot{\vee}_i^p$ with $i \in [1..N]$ is the *exclusive disjunction*, meaning that a signal $s(k)$ has to satisfy one subformula φ_i and not any other subformulae (mutually exclusive satisfaction). Formally, $s(k) \vDash \varphi \equiv s(k) \vDash \varphi_i \wedge s(k) \nvDash \varphi_j$, $i \neq j$. $\boxdot_I^w\varphi$ is the *exclusive* *always* capturing that formula φ has to be satisfied during the whole interval or not at all. Formally $s(k) \models \varphi \equiv (s(k') \models \varphi, \forall k' \in I) \vee (s(k') \not\models \varphi, \forall k' \in I)$. Finally, $\Diamond_I^w \varphi$ is the *exclusive eventually* that captures that a formula φ has to be satisfied at only one-time step and not anymore during the time interval I, formally, $s(k) \models \varphi \equiv s(k') \models \varphi \wedge s(k'') \not\models \varphi$, $k' \neq k''$, $\forall k', k'' \in I$.

Example 3. *Consider a quadrotor in a planar convex environment $\mathcal{M}$ with different regions of interest $\mathcal{R} = \{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}\} \subset \mathcal{M}$ that can be disjoint $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ or containing overlapping region $\mathcal{R}_i \cap \mathcal{R}_j \neq \emptyset$ with $\mathcal{R}_i \neq \mathcal{R}_j \in \mathcal{R}$. The solutions in Fig. 1(b) can be obtained by the following specifications using the exclusive operators. $\varphi_1 = \Box^w(\vee^p(\mathcal{A}, \mathcal{B}))$ with $p = [p_1, p_2]$, $p_1 > p_2$ and arbitrary w (purple dashed trajectory). $\varphi_2 = \Box^w(\wedge^p(\mathcal{A}, \mathcal{B}))$ with $p = [p_1, p_2]$, $p_1 = p_2$ and arbitrary w (red dashed trajectory). Lastly, the (green dashed trajectory) could be obtained with $\varphi_2 = \Box^w(\vee^{p_1}(\wedge^{p_2}(\mathcal{A}, \mathcal{B}), \mathcal{D}))$ with $p_1 = [p_{11}, p_{12}]$, $p_{11} < p_{12}$.*

B. System dynamics

The system dynamics are captured using linear difference equations as follows

$$s(k+1) = As(k) + Bu(k) + D, \quad s(0) = s_\circ, \tag{5}$$

where $s(k) \in S \subseteq \mathbb{R}^n$ is the state variable at time $k \in \mathbb{Z}_{\geq 0}$, S is the state space of the signals, $u(k) \in \mathbf{U} \subseteq \mathbb{R}^m$ is the control input, A and B are the state transition and input matrices of appropriate sizes, and D is the exogenous inputs or additive disturbances.

C. Customized partial satisfaction control synthesis problem

Our previous work [Cardona and Vasile(2023)] introduced the partial satisfaction problem over STL specifications. For this, we introduced a partial order considering the *depth*[1] in the AST, capturing the STL specification. Giving higher priority to the satisfaction of subformulae of minimal depth (i.e., closer to the root) guarantees maximal partial satisfaction. Although it was enough for synthesizing control inputs that satisfied as much as possible the mission specification when handling conflicting subformulae, it did not always lead to the best decision-making. The decision on what to satisfy was globally computed without providing preferences on what can or cannot be violated. However, this is not always desirable since there is no control over what matters the most for satisfaction (e.g., in case of conflict between a safety constraint and the system's behavior, safety constraints should be preferred over the desired behavior). Hence, in this work, we deal with the partial satisfaction problem over wSTL+ specifications where the weights p and w serve as a descriptor of preferences that guide and modulate the tie-breaking rules over satisfaction.

[1]The depth of a formula φ' with respect to a formula φ is the path distance between the root of φ's AST and the node associated with φ'. If the φ' is not a subformula of φ, the depth is by convention ∞.

Lastly, we define the *partial satisfaction robustness [Cardona and Vasile(2023)]*,

$$\varrho(s, \varphi) = \min_{(\varphi_i, k) \in F_\varphi(s)} \tilde{\rho}(s, \varphi_i, k), \tag{6}$$

where $F_\varphi(s) = \{(\varphi_i, k) \mid \nexists(\varphi_i', k') \text{ s.t. } \varphi_i \sqsubset \varphi_i', (s, k) \models \varphi_i, (s, k') \models \varphi_i'\}$ is the set of lowest-depth subformulae satisfied by s.

Problem 1. *Given a discrete linear system dynamics* (5)*, and a wSTL+ specification φ, find input signal $u(k)$ such that the generated state trajectory $s(k)$ satisfies φ as much as possible while considering the user preferences and maximizing the partial satisfaction robustness* (6) *over all time-horizon* $\|\varphi\|$ (3)*. Formally, we have a bi-level optimization problem*

$$\begin{aligned} \max_{\mathbf{u}} \quad & \varrho(s, \varphi) \\ \text{s.t.} \quad & \mathbf{u} \textit{ induces } s \\ & s \in \max_{\mathbf{u}'}{}^{\varphi}\{s'\} \ \text{ s.t. } \ \mathbf{u}' \textit{ induces } s' \end{aligned} \tag{7}$$

where $\max^\varphi\{s'\}$ denotes the trajectory s' that maximizes the satisfaction of φ (i.e., formula with minimal depth that can be satisfied while capturing user preferences).

Problem 1, takes a mission specification φ and synthesizes a signal control u that generates a state trajectory s that satisfies subformulae of lowest depth modulated by the weights p and w, and has the largest minimum robustness among them. The inner level finds the customized satisfaction of the lowest depth, while the outer level accounts for their robustness.

Remark 1. *Note that in case of conflicting subformulae, standard robustness for STL [Donzé and Maler(2010)], [Fainekos and Pappas(2009)] and wSTL [Mehdipour et al.(2020)Mehdipour, Vasile, and Belta] produces minimal violating solutions and partially deviated violation depending on the preference specified, respectively. However, none of the subformulae is chosen to be satisfied. In contrast, wSTL+ guarantees partial satisfaction by choosing subformulae based on user preferences captured by the weights.*

IV. Control Synthesis Encoding with Partial Satisfaction

In this section, we propose to solve Problem 1 in two steps that decouple the bi-level optimization problem. First, we propose a Mixed Integer Linear Programming (MILP) formulation to find the maximum satisfaction considering inclusive and exclusive preferences. The MILP corresponds to the inner optimization in (7). Second, we introduce a Linear Program (LP) that approximates the solution to the outer level of (7) using the solution of the inner level. The two-step approach trades off optimality with runtime performance.

A. MILP encoding of wSTL+ satisfaction fractions

In this section, we formulate Problem 1 as an optimization problem and introduce a Mixed Integer Linear Program (MILP) encoding for wSTL+. We consider the following assumptions throughout the rest of the paper.

Assumption 1. *wSTL+ specifications are over linear predicates.*

While wSTL+ formulae can be defined over general, non-linear predicates $l_\mu(s(k)) \geq \pi$ for some function $l : \mathbb{R}^n \to \mathbb{R}$, we limit them to simple linear functions $s_i(k) \geq \pi$ (or $s_i(k) \leq \pi$, see below). Our MILP encoding can still be employed by introducing output variables $y_\mu = l_\mu(s(k))$ for all non-linear predicates μ, and using piecewise-linear approximations of the output functions l_μ.

Assumption 2. *wSTL+ specifications are in positive normal form [Sadraddini and Belta(2015)].*

The assumption is not limiting because any STL, and by extension, the wSTL+ formula, can be put in positive normal form, where the negation operators are only in front of predicates. In the following, we eliminate negations by considering predicates defined with either $\geq$ or $\leq$ comparison operators.

The foundation of our encoding of wSTL+ formulae is based on the fraction of subformulae-time pairs that signals satisfy. Instead of encoding margins that are propagated towards a formula's root to compute robustness [Sadraddini and Belta(2015)], [Raman et al.(2014)Raman, Donzé, Maasoumy, Murray, Sangiovanni-Vincentelli, and Seshia],

Let φ be a wSTL+ formula specification. Let us consider that every weight p and w are normalized such that

$$p_i = \frac{p'_i}{\max_{j=1}^{N} p'_j}, \quad w_i = \frac{w'_i}{\max_{j=1}^{|\bar{I}|} w'_j}.$$

The MILP encoding is defined recursively over the nodes of the AST of wSTL+ formula φ starting from the leaves, the *predicates*.

Let $\mu := s(k) \geq \pi$ be a predicate. We define the variable $z_k^\mu \in \mathbb{B}$ that take value one if predicate μ is considered in the satisfaction of formula φ at time $k \in [0..\|\varphi\|]$, where $\|\varphi\|$ is the time horizon of φ computed based on (3). Let M be a large enough number (e.g., larger than the largest upper bound of signals used in wSTL+ specification φ). The following constraints capture the satisfaction of predicates

$$\varphi = \mu \Rightarrow \begin{cases} s(k) - \pi + M(1 - z_k^\mu) \geq 0 \\ s(k) - \pi - M z_k^\mu \leq 0 \end{cases} . \tag{8}$$

Inclusive preferences constraints: For all-inclusive operators in wSTL+, let us define $z_k^\varphi = [0,1]$ capturing the fraction of preferences satisfaction depending on the operator.

For *inclusive conjunction* operator z_k^φ takes value one if all subformulae φ_i with $i \in [1..N]$ are satisfied. Let us define $z_k^{\varphi_i} \in [0,1]$ capturing the fraction of satisfaction of subformula φ_i. The constraint capturing the inclusive conjunction semantics is

$$\varphi = \bigwedge_i^p \varphi_i \Rightarrow z_k^\varphi = \frac{\sum_i^N p_i z_k^{\varphi_i}}{\sum_i^N p_i}. \tag{9}$$

Note that as weights are normalized, the constraint of the inclusive conjunction takes the form of a weighted arithmetic mean. Taking value one only if all $z_k^{\varphi_i}$ are one, in case of conflicting subformulae, the one with the highest weight p is chosen.

For *inclusive disjunction* z_k^φ takes value one if at least one subformula ϕ_i is satisfied and the weight p_i is equal to one. Which captures the preference of choosing the subformula with the highest weight. The fraction of satisfaction of each subformula φ_i captured by $z_k^{\varphi_i} \in [0,1]$ The following constraint captures the inclusive disjunction semantics

$$\varphi = \bigvee_i^p \varphi_i \Rightarrow z_k^\varphi = \max_{i=1:N}\{z_k^{\varphi_i} \cdot p_i\}. \tag{10}$$

However, in contrast, as in [Cardona and Vasile(2023)] z_k^φ can be lower than one without implying violation; instead, it captures that the subformula chosen was not the most preferred one.

For the *inclusive always* operator, we consider the same logic as for *inclusive conjunction* but with weight and variables over time rather than over subformulae $z_{k'}^\psi \in [0,1]$. Then we have

$$\varphi = \Box_I^w \psi \Rightarrow z_k^\varphi = \frac{\sum_{k' \in k+I} z_{k'}^\psi w_{k'}}{\sum_{k' \in k+I} w_{k'}}. \tag{11}$$

The *inclusive eventually* operator follows the same as the *inclusive disjunction* operator but again considers time instead of subformulae for the weights and variables $z_{k'}^\psi \in [0,1]$. Encoded as follows

$$\varphi = \Diamond_I^w \psi \Rightarrow z_k^\varphi = \max_{k' \in k+I}\{z_{k'}^\psi w_{k'}\}. \tag{12}$$

Exclusive preferences constraints: For the *exclusive conjunction* is the same constraint as for *inclusive disjunction* $z_k^{\varphi_i} \in [0,1]$, but instead of $z_k^\varphi \in [0,1]$ we consider $z_k^\varphi \in \mathbb{B}$ notice that this capture that either all subformulae φ_i are satisfied, or not at all.

$$\varphi = \dot{\bigwedge_i}^p \varphi_i \Rightarrow z_k^\varphi = \frac{\sum_i^N p_i z_k^{\varphi_i}}{\sum_i^N p_i}. \tag{13}$$

For *exclusive disjunction*, we use the same constraint used for the *inclusive conjunction*, also considering $z_k^\varphi = [0,1]$ and $z_k^{\varphi_i} \in [0,1]$. Let us define an auxiliary variable $b_i \in \mathbb{B}$ taking value one if $z_k^{\varphi_i}$ is greater than zero. The following set of constraints captures the semantics of *exclusive disjunction*

$$\varphi = \dot{\bigvee_i}^p \varphi_i \Rightarrow \begin{cases} z_k^\varphi = \max_{i=1:N}\{z_k^{\varphi_i} \cdot p_i\} \\ z_k^{\varphi_i} \leq b_i \\ \sum_{i=1}^N b_i \leq 1 \end{cases} . \tag{14}$$

Note that the first constraint captures that at least one subformula has to be satisfied, preferably the one with the maximum weight. Furthermore, the last two capture that only one subformula has to be satisfied and not the rest, generating the desired mutual exclusivity between subformulae.

The *exclusive always* is captured in a similar way to *exclusive conjunction* but considering time weights and variables instead as subformulae $z_{k'}^\psi \in [0,1]$. Also, with $z_k^\varphi = \{0,1\}$ capturing that subformula $\psi \sqsubseteq \varphi$ is either fully satisfied or not considered at all in the satisfaction.

$$\varphi = \boxdot_I^w \psi \Rightarrow z_k^\varphi = \frac{\sum_{k' \in k+I} z_{k'}^\psi w_{k'}}{\sum_{k' \in k+I} w_{k'}}. \tag{15}$$

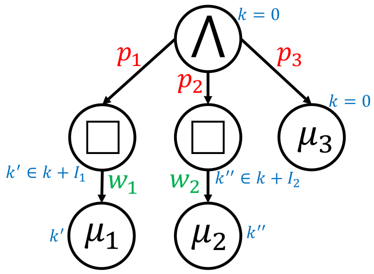


Fig. 2. Partial satisfaction modulation with weights values.

The *exclusive eventually*, is captured similarly as *exclusive disjunction* but considering time weights and variables $z^{\psi}_{k'} \in [0,1]$ and $z^{\varphi}_k \in [0,1]$, we also consider an auxiliary variable $b_i \in \mathbb{B}$ with $i \in [\underline{I}..\bar{I}]$

$$\varphi = \diamondsuit^w_I \ \psi \Rightarrow \begin{cases} z^{\varphi}_k = \max_{k' \in k+I}\{z^{\psi}_{k'} w_{k'}\} \\ z^{\varphi}_{k'} \le b_i \\ \sum_{i=1}^{\bar{I}} b_i \le 1 \end{cases} . \qquad (16)$$

Capturing that subformula, $\psi \sqsubset \varphi$ is satisfied at only one-time step and not at any other time step within the time interval. Making a mutual exclusivity at every time step within the time interval.

Finally, the inner level of the optimization problem is formulated as follows

$$\begin{aligned} \max_{s,u,z} \quad & z^{\varphi}_0 \\ \text{s.t.} \quad & (5) \quad \text{(linear dynamics)} \\ & (8)-(16) \quad \text{(mission satisfaction)} \end{aligned}$$

where z^{φ}_0 is the root node in the AST of the wSTL+ specification φ.

Remark 2. *Any wSTL+ formula can be represented using an Abstract Syntax Tree (AST) in which intermediate nodes correspond to logical and temporal operators and leaves to predicates [Hopcroft et al.(2001)Hopcroft, Motwani, and Ullman]. Our MILP encoding relies on a recursion definition that uses the AST to compute the overall fractions of satisfaction of the specification formula.*

B. Partial satisfaction robustness LP

In this section, we propose to approximate the robust solution of Problem 1 using an LP based on solutions of the MILP (inner optimization level). Let $\{z^{\varphi'}_k\}_{\varphi' \sqsubseteq \varphi, k \in [0..\|\varphi\|]}$ be the set of solution decision variables for satisfaction of φ from (IV-A). The following LP,

$$\begin{aligned} \max_{s,u} \quad & \tilde{\rho}(\varphi, s, k) \\ \text{s.t. } (5), \ & \tilde{\rho}(\varphi, s, k) \le s_i(k) - \mu, \forall \mu \text{ with } z^{\mu}_k = 1 \end{aligned}, \qquad (17)$$

computes the signal s and control u that maximize the robustness of all predicates μ at all times k that are satisfied in the reference solution encoded by z^{φ}_k.

C. Partial satisfaction encoding Analysis

In this section, we remark on the properties of the MILP encoding of wSTL+. First, if a wSTL+ specification φ and all its subformulae $\varphi_i \sqsubseteq \varphi$ are not satisfied by a trajectory $\mathbf{s} = s(0), s(1), \ldots, s(\|\varphi\|)$, therefore $z^{\varphi}_0 = 0$ (root node in the AST). The property follows by structural induction over the entire AST as $\mathbf{s} \not\models \varphi_i$ for every $\varphi_i \in \varphi$. Then every leave and intermediate node variables $z^{\varphi_i}_k = 0$ in the AST with $k = [0..\|\varphi\|]$, thus $z^{\varphi}_0 = 0$.

Second, in our previous work [Cardona and Vasile(2023)], we introduced the encoding as fractions of satisfaction, then the value obtained after the optimization of z^{φ}_0 indicates the fraction of satisfaction of the wSTL+ specification φ. The same applies for any subformula $\varphi \sqsubseteq \varphi$, the value of $z^{\varphi_i}_k$ indicates the fraction of satisfaction of φ depending on every subformula φ' with greater depth in the AST such that $\varphi' \sqsubset \varphi_i$. This property of the fraction of satisfaction no longer holds in our MILP encoding for wSTL+. In contrast, the value of z^{φ}_0 indicates how close the induced trajectory $\mathbf{s}$ satisfying φ to capture all the user preferences indicated in the specification. This is given by the fact that in the case of *disjunctions* and *eventually* inclusive and exclusive operators, if the subformula φ_i counted for satisfaction is not the one or one with the maximum weight p or w then the resulting variable $z^{\varphi}_k < 1$ which propagates back to the root making $z^{\varphi}_0 < 1$. Note that this does not indicate a violation of the specification. It only captures that the trajectory chosen is not the most preferred one.

Thirdly, as a consequence of the previous properties, it trivially follows by contradiction that $z^{\varphi}_0 = 1$ iff the trajectory induced $\mathbf{s}$ is in the set of the most preferred solutions that capture all preferences and priorities indicated in the wSTL+ specification φ. Therefore the trajectory fully satisfies the specification $\mathbf{s} \models \varphi$.

Next, we want to remark that the weights directly modulate the satisfaction of the wSTL+ specification capturing user preferences and priorities. Let us consider the following mission specification $\varphi = \wedge^p(\square^{w_1}(\mu_1), \square^{w_2}(\mu_2), \mu_3)$, we consider μ_1, μ_2, and μ_3 conflicting predicates. The AST that captures the specification is shown in Fig. 2 consider $w_1 = \mathbf{1}_{k'}$ and $w_2 = \mathbf{1}_{k''}$, therefore we can modulate which predicate to satisfy. For instance, $p_1 = p_2$ and $p_3 > p_1$ make μ_3 to be accounted into satisfaction and not the other two. On the other hand, if $p_3 < p_1$, μ_1 is added to the satisfaction, and the other two are ignored.

Finally, the solution of LP (17) has maximum PS robustness $\varrho(s, \varphi)$ over all trajectories that satisfy the same set of subformulae (φ', k) as s^*. Note that an optimal solution to Problem III might satisfy other subformulae than s^* even if both achieve the same number at each depth. As such, the LP (17) may lead to suboptimal solutions.

V. Case Studies

In this section, we showcase and test the functionality of the wSTL+ MILP encoding under partial satisfaction conditions (PS-wSTL+). First, we show the versatility of the exclusive operators added to the wSTL+ language specification. Then, we show the control synthesis for an agent satisfying a specification of navigating in a planar environment with temporal and logic constraints, modulating the partial satisfaction solution through the weights. Lastly, we show the time performance and complexity comparison between

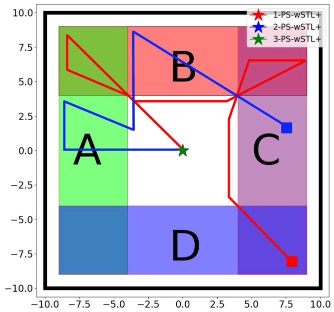


Fig. 3. Exclusive operators functionality.

STL, PS-STL, and PS-wSTL+ under partial satisfaction MILP solutions. All computations in the case studies were performed on a PC with 20 cores at 3.7 GHz with 64 GB of RAM. We used Gurobi [Gurobi Optimization, LLC(2021)] as the MILP solver.

A. Exclusive operators functionality

Let us consider a single robot navigating in a planar environment $\mathcal{M} \subset \mathbb{R}^2$ shown in Fig. 3. Regions of interest $\mathcal{R} = \{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}\}$, with $\mathcal{A} = [-4,-9] \times [-9,9]$, $\mathcal{B} = [-9,4] \times [9,9]$, $\mathcal{C} = [4,-9] \times [9,9]$, and , $\mathcal{D} = [-9,-4] \times [9,-9]$. Note that some regions are not disjoint, e.g., $\mathcal{A} \cap \mathcal{B} \neq \emptyset$, or disjoint e.g., $\mathcal{A} \cap \mathcal{C} = \emptyset$. We arbitrarily choose $A = B = I_{2\times 2}$, for the robot dynamics as in (5). Thus, $s(k) = [s_x(k), s_y(k)]^\top \in \mathbb{R}^2$. We consider initial position as $s(0) = (s_x(0), s_y(0)) = (0,0)$. We consider the following wSTL+ specifications

$$\begin{aligned}
\varphi_1 &= \barwedge^{p_1}\left(\boxdot^w_{I_1}(\barwedge^{p_1}(\mathcal{A},\mathcal{B})), \boxdot^w_{I_2}(\barwedge^{p_1}(\mathcal{B},\mathcal{C})), \boxdot^w_{I_3}(\barwedge^{p_1}(\mathcal{C},\mathcal{D}))\right),\\
\varphi_2 &= \barwedge^{w}\left(\boxdot^w_{I_1}(\veebar^{p}(\mathcal{A},\mathcal{B},\mathcal{D})), \boxdot^w_{I_2}(\veebar^{p}(\mathcal{B},\mathcal{C},\mathcal{A})),\right.\\
&\qquad \left.\boxdot^w_{I_2}(\veebar^{p}(\mathcal{C},\mathcal{D},\mathcal{A}))\right),
\end{aligned}$$

where $I_1 = [5..9]$, $I_2 = [14..18]$, $I_3 = [24..26]$, $p_1 = [1,1]$ all weights w are defined to be one of appropriate size, and $p = [1, 0.5, 0.5]$. For simplicity, with a slight abuse of notation, we define the formulae directly over the regions instead of defining predicates over all four boundaries of each region.

In Fig. 3. we show the solution of both specifications. Note that φ_1 (red trajectory) uses exclusive conjunctions for specifying the regions that must be visited within the intervals. As this operator impose that all subformulae have to be satisfied or not at all, the only solution possible is given when the robot visits the overlapping regions between areas of interest requested. On the other hand, φ_2 (blue trajectory) uses exclusive disjunction, and the weight p assigns a larger preference to the first region specified the robot visits that region without crossing the other regions. Since exclusive disjunction imposes one subformula and not any other subformulae. Lastly, we show the importance of declaring exclusive operators correctly. We modify φ_1, defining exclusive conjunctions between disjoints regions, and therefore the solution is set to not visit any region (green start) since visiting both regions simultaneously is physically impossible.

B. Control synthesis

Let us consider the same robot defined in the previous section, with the initial position as $s(0) = (s_x(0), s_y(0)) = (-9,-9)$. We define disjoint regions of interest $\mathcal{A} = [-9.5,-5.5]^2$, $\mathcal{B} = [5.5, 9.5] \times [-9.5, -5.5]$, $\mathcal{C} = [5.5, 9.5]^2$, $\mathcal{D} = [-9.5,-5.5] \times [5.5, 9.5]$ in $\mathcal{M}$, and region $\mathcal{E} = [-2.5, 2.5]^2 \subseteq \mathcal{M}$ that robot $s(k)$ needs to avoid. We consider the STL ϕ and wSTL φ specifications

$$\phi = (\Box_{[0,1]}\mathcal{A}) \wedge (\Box_{[10,15]}\mathcal{C}) \wedge (\Box_{[25,30]}\mathcal{D}) \wedge (\Box_{[0,30]}\mathcal{E}^c), \tag{18}$$

$$\varphi = \wedge^p\left((\Box^w_{[0,1]}\mathcal{A}), (\Box^w_{[10,15]}\mathcal{C}), (\Box^w_{[25,30]}\mathcal{D}), (\Box^w_{[0,30]}\mathcal{E}^c)\right), \tag{19}$$

Going to regions of interest can be specified in STL and wSTL+ form by constraining s_x and s_y inside the region boundaries. Note that we use $\mathcal{E}^c$, which indicates that the robot is free to move in any location out of this area.

In Fig. 4(a), we set all weights w and p to one. It can be seen that the wSTL+ (green dashed line) and PS-STL [Cardona and Vasile(2023)] (blue dashed line) have similar trajectories for satisfying the specification, and all three encodings including STL [Sadraddini and Belta(2015)] (solid red line) can satisfy the mission.

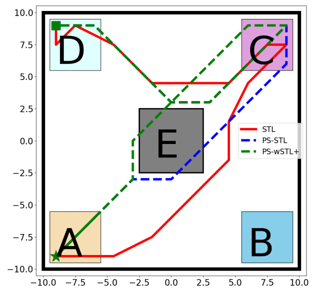
In Fig.4(b), we show how the solution for the wSTL+ (green dashed line) can change by using exclusive temporal and logical operators instead of standard operators specifically for avoiding region $\mathcal{E}$ the trajectory is carefully synthesized never to cross it.

In Fig.4(c), we show a case where there is a conflict between subformulae to visit two regions simultaneously which are physically impossible, specified in STL and wSTL+ as $\phi = \Box_{[5,6]}(\mathcal{B} \wedge \mathcal{D})$, $\varphi = \Box^w_{[5,6]}(\wedge^p(\mathcal{B}, \mathcal{D}))$, where $w = [1,1]$ and $p = [0.5, 1]$. As expected, standard STL encoding [Sadraddini and Belta(2015)] minimally violates both subformulae. By using the *hierarchical method* in PS-STL [Cardona and Vasile(2023)] the solution computed is going to region $\mathcal{B}$. However, this solution is obtained because it is the first specified. It might not be the most preferred option. In contrast, we specify a preference to visit region $\mathcal{D}$ by defining a larger weight, and the solution captures the user preference, and the robot visits the desired region.

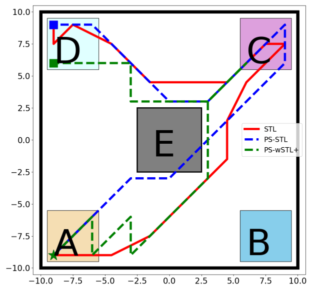
Lastly, let us consider the initial position of the agent as $s_x(0) = -5$, $s_y(0) = -1$, and the following wSTL specification $\varphi = \wedge^{p_i}\left((\Box^w_{[0,7]}\mathcal{E}^c), (\Box^w_{[8,18]}\mathcal{C})\right)$, see Fig. 4(d). We generate different trajectories, and the first four are constrained by control inputs bounds of two units. Some of them are getting closer or farther to $\mathcal{E}$ according to the weights. However, for the last case, the control bounds are three units, and the solution is to go around the bottom boundary of $\mathcal{E}$, keep distance to the obstacle, and reach maximum robustness. Thus, varying weights may produce topologically similar trajectories or different from the STL one.

C. Time performance and complexity comparison

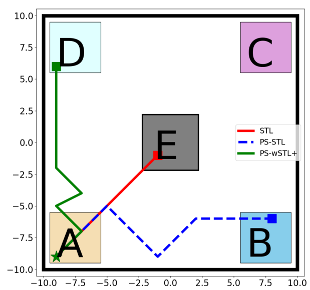
We show the run time performance comparison between STL [Sadraddini and Belta(2015)], partial satisfaction for STL (PS-STL) [Cardona and Vasile(2023)], and our partial satisfaction with wSTL (PS-wSTL) encodings with random weights by gradually increasing the size of the mission specification. Let us consider six variables x, y, z, u, v, and



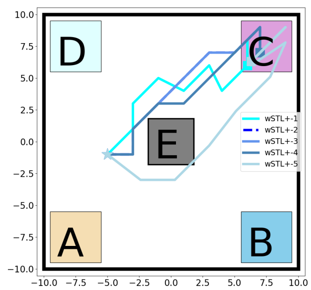
(a) Feasible symmetric specification for PS-STL and PS-wSTL+ (unit weights and inclusive operators).



(b) Feasible symmetric specification for PS-STL and PS-wSTL+ (unit weights and exclusive operators).



(c) Partial satisfaction choosing the most preferred subformula.



(d) Trajectories around an obstacle changing the weights in the wSTL specification

Fig. 4. Trajectories for wSTL φ and STL ϕ specifications in a two-dimensional environment.

w, all with a lower-bound of -9 and upper-bound of 9. The STL ϕ and wSTL φ specification are the following

$$\phi = \bigwedge_{1}^{n} \mathcal{T}_I \left((\mathfrak{s}_1 \otimes_1 \Xi_1) \,\mathcal{L}\, (\mathfrak{s}_2 \otimes_2 \Xi_2) \right),$$

$$\varphi = \bigwedge_{1}^{n\; p} \tilde{\mathcal{T}}_I^{w} \left(\tilde{\mathcal{L}}^{p} \left((\mathfrak{s}_1 \otimes_1 \Xi_1), (\mathfrak{s}_2 \otimes_2 \Xi_2) \right) \right),$$

where $\mathcal{T} \in \{\Box, \Diamond\}$, $\tilde{\mathcal{T}} \in \{\Box, \Diamond, \boxdot, \dot{\Diamond}\}$, $\mathcal{L} \in \{\wedge, \vee\}$, $\tilde{\mathcal{L}} \in \{\wedge, \vee, \dot{\vee}, \dot{\wedge}\}$, $\mathfrak{s}_1$ and $\mathfrak{s}_2 \in \{x, y, z, u, v, w\}$, $\otimes \in \{<, \leq, >, \geq\}$, Ξ_1 and $\Xi_2 = \text{rand}(-8, 8)$ are variables randomly chosen, n is an iterator that grows from 1 to 200, and the time interval of the temporal operator is defined randomly as $I = [n + 4, \ldots, n + 4 + \text{rand}(1, 5)]$, and the weights for Boolean p and temporal w operators are randomly chosen in an interval $(0, 1]$. In Fig. 5, we compare time performance for STL, PS-STL, and PS-wSTL+ growing formulae with random weights. Note that the STL performance grows linearly and is faster than the other two. However, there is a slight difference between STL and PS-wSTL+, which is expected since more constraints are required in the encoding to capture the importance or preferences in the specification. Moreover, no partial satisfaction is imposed by the STL solution. The performance of PS-wSTL+ is better than PS-STL. Both capture partial satisfiability, but only PS-wSTL+ captures preferences and importance. We hypothesize that weights act as tie-breakers, allowing the MILP solver to prune sub-optimal infeasible solutions.

VI. Conclusions

We extended wSTL with exclusive logical and exclusive temporal operators denoted as wSTL+. We presented a

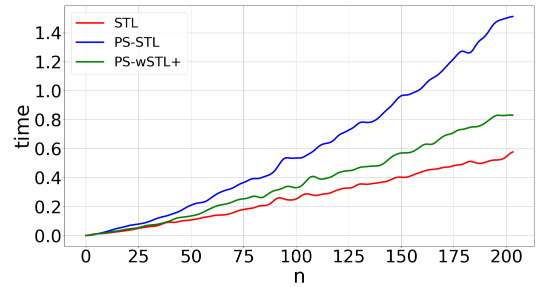


Fig. 5. Time performance between random generated specification in STL [Sadraddini and Belta(2015)], PS-STL [Cardona and Vasile(2023)], PS-wSTL+.

Mixed Integer Linear Programming formulation for wSTL+ that accepts partial satisfaction solutions in case of conflicting or infeasible subformulae. Partial satisfaction is achieved by a bilevel optimization problem where the inner level maximizes partial satisfaction via fractions of satisfaction with weights modulated by user preferences. The outer level maximizes the robustness of satisfiable subformulae obtained in the inner level. Finally, the time performance of the MILP encoding is shown compared with standard STL and partial satisfaction encoding of STL. Showing there is a small cost to pay at the expense of being able to specify preferences and importance compared to STL, but there is a performance improvement with respect to PS-STL.

References

[Baier and Katoen(2008)] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

[Bellini et al.(2000)Bellini, Mattolini, and Nesi] Pierfrancesco Bellini, Riccardo Mattolini, and Paolo Nesi. Temporal logics for real-time system specification. *ACM Computing Surveys (CSUR)*, 32(1):12–42, 2000.

[Bisoffi and Dimarogonas(2020)] Andrea Bisoffi and Dimos V Dimarogonas. Satisfaction of linear temporal logic specifications through recurrence tools for hybrid systems. *IEEE Transactions on Automatic Control*, 66(2):818–825, 2020.

[Cai et al.(2020a)Cai, Peng, Li, Gao, and Kan] Mingyu Cai, Hao Peng, Zhijun Li, Hongbo Gao, and Zhen Kan. Receding horizon control-based motion planning with partially infeasible ltl constraints. *IEEE Control Systems Letters*, 5(4):1279–1284, 2020a.

[Cai et al.(2020b)Cai, Peng, Li, and Kan] Mingyu Cai, Hao Peng, Zhijun Li, and Zhen Kan. Learning-based probabilistic ltl motion planning with environment and motion uncertainties. *IEEE Transactions on Automatic Control*, 66(5):2386–2392, 2020b.

[Cardona and Vasile(2023)] Gustavo A. Cardona and Cristian-Ioan Vasile. Partial satisfaction of signal temporal logic specifications for coordination of multi-robot systems. In *Algorithmic Foundations of Robotics XV*, pages 223–238, Cham, 2023. Springer International Publishing. ISBN 978-3-031-21090-7.

[Choudhury et al.(2020)Choudhury, Gupta, Kochenderfer, Sadigh, and Bohg] Shushman Choudhury, Jayesh K Gupta, Mykel J Kochenderfer, Dorsa Sadigh, and Jeannette Bohg. Dynamic multi-robot task allocation under uncertainty and temporal constraints. *Robotics: Science and Systems*, 2020.

[Dokhanchi et al.(2014)Dokhanchi, Hoxha, and Fainekos] Adel Dokhanchi, Bardh Hoxha, and Georgios Fainekos. *5th International Conference on Runtime Verification, Toronto, ON, Canada.*, chapter On-Line Monitoring for Temporal Logic Robustness, pages 231–246. Springer, 2014. ISBN 978-3-319-11164-3.

[Donzé and Maler(2010)] Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 92–106. Springer, 2010.

[Fainekos and Pappas(2009)] Georgios E Fainekos and George J Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410(42):4262–4291, 2009.

[Guo and Dimarogonas(2015)] Meng Guo and Dimos V Dimarogonas. Multi-agent plan reconfiguration under local ltl specifications. *The International Journal of Robotics Research*, 34(2):218–235, 2015.

[Gurobi Optimization, LLC(2021)] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2021. URL https://www.gurobi.com.

[Haghighi et al.(2019)Haghighi, Mehdipour, Bartocci, and Belta] Iman Haghighi, Noushin Mehdipour, Ezio Bartocci, and Calin Belta. Control from signal temporal logic specifications with smooth cumulative quantitative semantics. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 4361–4366. IEEE, 2019.

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

[Kamale et al.(2021)Kamale, Karyofylli, and Vasile] Disha Kamale, Eleni Karyofylli, and Cristian-Ioan Vasile. Automata-based optimal planning with relaxed specifications. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021.

[Kress-Gazit et al.(2018)Kress-Gazit, Lahijanian, and Raman] Hadas Kress-Gazit, Morteza Lahijanian, and Vasumathi Raman. Synthesis for robots: Guarantees and feedback for robot behavior. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:211–236, 2018.

[Lacerda et al.(2015)Lacerda, Parker, and Hawes] Bruno Lacerda, David Parker, and Nick Hawes. Optimal policy generation for partially satisfiable co-safe ltl specifications. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[Lahijanian et al.(2015)Lahijanian, Almagor, Fried, Kavraki, and Vardi] Morteza Lahijanian, Shaull Almagor, Dror Fried, Lydia E Kavraki, and Moshe Y Vardi. This time the robot settles for a cost: A quantitative approach to temporal logic planning with partial satisfaction. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[Leahy et al.(2021)Leahy, Serlin, Vasile, Schoer, Jones, Tron, and Belta] Kevin Leahy, Zachary Serlin, Cristian-Ioan Vasile, Andrew Schoer, Austin M Jones, Roberto Tron, and Calin Belta. Scalable and robust algorithms for task-based coordination from high-level specifications (scratches). *IEEE Transactions on Robotics*, 2021.

[Lindemann and Dimarogonas(2019a)] Lars Lindemann and Dimos V Dimarogonas. Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks. *IEEE control systems letters*, 3(3):757–762, 2019a.

[Lindemann and Dimarogonas(2019b)] Lars Lindemann and Dimos V Dimarogonas. Robust control for signal temporal logic specifications using discrete average space robustness. *Automatica*, 101:377–387, 2019b.

[Maler and Nickovic(2004)] Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 152–166. Springer, 2004.

[Mehdipour et al.(2020)Mehdipour, Vasile, and Belta] Noushin Mehdipour, Cristian-Ioan Vasile, and Calin Belta. Specifying user preferences using weighted signal temporal logic. *IEEE Control Systems Letters*, 5(6):2006–2011, 2020.

[Raman et al.(2014)Raman, Donzé, Maasoumy, Murray, Sangiovanni-Vincentelli, and Seshia] Vasumathi Raman, Alexandre Donzé, Mehdi Maasoumy, Richard M Murray, Alberto Sangiovanni-Vincentelli, and Sanjit A Seshia. Model predictive control with signal temporal logic specifications. In *53rd IEEE Conference on Decision and Control*, pages 81–87. IEEE, 2014.

[Sadraddini and Belta(2015)] Sadra Sadraddini and Calin Belta. Robust temporal logic model predictive control. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 772–779. IEEE, 2015.

[Sun et al.(2022)Sun, Chen, Mitra, and Fan] Dawei Sun, Jingkai Chen, Sayan Mitra, and Chuchu Fan. Multi-agent motion planning from signal temporal logic specifications. *IEEE Robotics and Automation Letters*, 7(2):3451–3458, 2022.