Optimal LTL$_f$ Synthesis*

Yujian Cao[1][0009−0008−0109−3049], Sven Schewe[1][0000−0002−9093−9518], Qiyi Tang[1][0000−0002−9265−3011], and Shufang Zhu[1][0000−0002−5922−8750]

University of Liverpool, Liverpool, UK
{yujian.cao, sven.schewe, qiyi.tang, shufang.zhu}@liverpool.ac.uk

Abstract. Strategy synthesis typically follows an all-or-nothing paradigm, returning unrealisable whenever a specification cannot be guaranteed in an uncertain environment. In this paper, we introduce optimal LTL$_f$ synthesis, where the goal is to realise as many objectives as possible from a given specification consisting of multiple objectives, especially for the case that they are not all jointly realisable. We first consider max-guarantee synthesis, which commits to a maximal set of objectives that we can a priori guarantee to realise. We then introduce max-observation synthesis, which maximises a posteriori realised objectives that may be incomparable on different executions. Finally, we present incremental max-observation synthesis, which further improves strategies by exploiting opportunities for stronger guarantees when they arise during an execution. Experimental results show that different variations of optimal synthesis scale broadly equally well, solving a large fraction of the benchmark instances within the given timeout, demonstrating the practical feasibility of the approach.

Keywords: LTL$_f$· Reactive Synthesis

1 Problem Setting

Consider an agent where we have five goals of equal value. The robot first chooses either a path that allows it to *surely* visit Room 1, or to go down a path where, depending on a choice of the environment, it can guarantee *either* to visit Rooms 2 and 3 *or* to visit Rooms 4 and 5.

If we only value guarantees that we can give *a priori*, then we would favour being sure to visit Room 1. We refer to this as *max-guarantee synthesis*: max-guarantee synthesis provides the maximal (or maximally valued) subset of objectives that can be jointly realised in the classic sense.

If we want to maximise (the value of) the specifications that we will achieve, on the other hand, we would prefer to satisfy two goals, even if we cannot commit to any of them in particular. We introduce *max-observation* synthesis to address this problem. Instead of committing to a fixed subset of objectives in advance,

* This work will be presented at IJCAI 2026. The full version is available at https://arxiv.org/abs/2605.11544

max-observation synthesis evaluates a strategy based on the objectives it actually satisfies along each execution, and focuses on maximising the number (or value) of achieved objectives.

We further improve max-observation synthesis to *incremental* max-observation synthesis, where we want to maximise the ensured number (value) of objectives for every history. In the robot example, the robot might find itself in a situation where, after visiting Room 2, it can choose to continue to Room 3, or go down a route where it can visit Rooms 4 and 5. If we only consider what the strategy can ensure from the beginning of its execution, these routes are of equal value, because *other* runs of the robot curtail this value to 2.

In incremental max-observation synthesis, we want more: after each observed history, we seek a strategy that maximises the value that can be ensured from that point onward. In this setting, we would prefer visiting Rooms 4 and 5 over visiting Room 3 only, since—for the given history—this allows us to increase our promise to visit at least three rooms overall.

2 Optimal LTL$_f$ Synthesis

Let Φ be a set of LTL$_f$ formulas over the alphabet $2^{\mathcal{X}\cup\mathcal{Y}}$, where $\mathcal{X}$ and $\mathcal{Y}$ are a fixed partition of the atoms into environment input and agent output, respectively. Let $\mathcal{P} = (\Phi, \mathcal{X}, \mathcal{Y})$ be a tuple and $G, V : \Phi \to (0, 1]$ be functions specifying the preference value of every LTL$_f$ formula $\varphi \in \Phi$ as guarantee (G) and observed value (V), respectively.

Max-guarantee synthesis. For a strategy $\sigma : (2^{\mathcal{X}})^* \to 2^{\mathcal{Y}}$, its guarantee value is $G(\sigma) = \sum_{\varphi\in\Phi:\sigma\triangleright\varphi} G(\varphi)$. A strategy σ^* is a *max-guarantee winning strategy* for $\mathcal{P}$ if $G(\sigma^*) = \max_\sigma G(\sigma)$.

Max-observation synthesis. For a trace $\pi \in (2^{\mathcal{X}\cup\mathcal{Y}})^\omega$, its observed value is $V(\pi) = \sum_{\varphi\in\Phi:\pi\models\varphi} V(\varphi)$, and the observed value of a strategy σ is $V(\sigma) = \min_{\mathbf{X}\in(2^{\mathcal{X}})^\omega} V(\pi(\mathbf{X}, \sigma))$. A strategy σ^* is a *max-observation winning strategy* for $\mathcal{P}$ if $V(\sigma^*) = \max_\sigma V(\sigma)$.

Incremental Max-observation synthesis. After a prefix $h \in (2^{\mathcal{X}\cup\mathcal{Y}})^*$, we might be in a position to provide *better* ensured values. Let $\mathsf{pre}(\sigma) \subseteq (2^{\mathcal{X}\cup\mathcal{Y}})^*$ be the set of prefixes of runs induced by σ, and $\mathsf{comp}(h) = \{\sigma \mid h \in \mathsf{pre}(\sigma)\}$ be the strategies compatible with h. For $h \in \mathsf{pre}(\sigma)$, we define $V_h(\sigma) = \min\{V(h \cdot \rho) \mid h \cdot \rho \in \pi(\mathbf{X}, \sigma),\ \mathbf{X} \in (2^{\mathcal{X}})^\omega\}$. A strategy σ^* is *incremental observation optimal* for $\mathcal{P}$ if $V_h(\sigma^*) = \max_{\sigma\in\mathrm{comp}(h)} V_h(\sigma)$ holds for all histories $h \in \mathrm{pre}(\sigma^*)$.

All three problems are 2EXPTIME-complete.

3 Solution

All three problems can be solved by reducing them to reachability games on a product automaton. They share the same game arena construction and differ in the accepting sets. For a reachability game $\mathcal{G} = (\mathcal{A}, F)$, we denote by $\mathrm{Win}(\mathcal{G})$ the set of states from which the agent has a winning strategy.

For every LTL$_f$ formula $\varphi_i \in \Phi$, we build its corresponding DFA $\mathcal{A}^i = (2^{\mathcal{X}\cup\mathcal{Y}}, \mathcal{Q}^i, q_0^i, \delta^i, F^i)$, where $F^i \subseteq \mathcal{Q}^i$ is the set of accepting states for φ_i. We then build the product automaton $\mathcal{A} = \bigotimes_{\varphi_i\in\Phi} \mathcal{A}^i$, such that $\mathcal{A} = (2^{\mathcal{X}\cup\mathcal{Y}}, \mathcal{Q}, q_0, \delta, \emptyset)$, where $\mathcal{Q} = \times_{\varphi_i\in\Phi}\mathcal{Q}^i$ is the product state space, $q_0 = (q_0^i)_{\varphi_i\in\Phi}$ is the initial state, δ is the product transition function induced by the $\{\delta^i\}_{\varphi_i\in\Phi}$. For each $\varphi_i \in \Phi$, we define $F^i_\otimes = \{(q_1, \cdots, q_n) \in \mathcal{Q} \mid q_i \in F^i\}$ as the set of states in the product automaton where φ_i is among the satisfied specifications.

Max-guarantee. We define $F_\Psi = \bigcap_{\varphi_i\in\Psi} F^i_\otimes$ for $\Psi \subseteq \Phi$. We iterate over subsets Ψ in non-increasing order of $\sum_{\varphi\in\Psi} G(\varphi)$ and solve $\mathcal{G}_\Psi = (\mathcal{A}, F_\Psi)$ until $q_0 \in \mathrm{Win}(\mathcal{G}_\Psi)$, and return a winning strategy.

Max-observation. We consider the finite set of values $v_1 > v_2 > \cdots > v_\ell$ that $\sum_{\varphi\in\Psi} V(\varphi)$ can take for subsets Ψ of Φ, and define $F_v = \bigcup\{F_\Psi \mid \sum_{\varphi\in\Psi} V(\varphi) \geq v\}$ for each of these values. We iterate over the value v in descending order and solve $\mathcal{G}_v = (\mathcal{A}, F_v)$ until $q_0 \in \mathrm{Win}(\mathcal{G}_v)$, and return a winning strategy.

Incremental max-observation. We can adapt the basic algorithm for max-observation. After solving each $\mathcal{G}_v$, we record for every state q the maximal value v_q together with the corresponding local strategy $\sigma_q \in 2^{\mathcal{Y}}$, and combine these local strategies into a global strategy.

This adaptation is inefficient: each state $q \in \mathrm{Win}(\mathcal{G}_{v_i})$ also belongs to $\mathrm{Win}(\mathcal{G}_{v_j})$ for all $j > i$, so each state is considered multiple times. We exploit this monotonicity by an improved algorithm: we grow the target by accumulating previous winning regions, so each state needs to be considered only once.

4 Experimental Evaluation

We evaluate our approach on 1145 benchmark instances taken from the LTL$_f$ synthesis track of SyntCOMP (`https://www.syntcomp.org/`). We restrict our evaluation to instances that are unrealisable and conjunction-decomposable, i.e. specifications given as a conjunction of multiple individual LTL$_f$ formulas, since our goal is to synthesise strategies that satisfy as many individual specifications as possible.

Generally, different methods work broadly equally fast. Within the 300s timeout, approximately 90% of the benchmark instances are solved, demonstrating the practical feasibility of our optimal LTL$_f$ synthesis techniques. The experimental results indicate that extending basic max-observation to incremental strategies incurs only minor overhead, and in some cases even improves performance.

An interesting observation is that, the extension of basic algorithm for max-observation can sometimes outperform the improved algorithm in runtime. This is because the improved algorithm works on larger target sets in each game iteration, which can reduce the number of iterations but makes each step more expensive in practice.