**Cost-Aware Path Planning under Co-Safe Temporal Logic Specifications: Supplementary Material**

Kyunghoon Cho, Junghun Suh, Claire J. Tomlin, and Songhwai Oh

In this supplementary material, we provide full proofs for probabilistic completeness and asymptotic optimality in Section I. In Section II, additional simulation results are provided on the convergence of the proposed method using more complex examples, the effects of different parameter configurations, and the use of a traveled-distance based cost function.

### I. Analysis

In this section, we prove the probabilistic completeness and asymptotic optimality of the proposed algorithm (CARL). For simplicity, proofs are made under the following conditions.

1. Each weight of high-level state \(w(q,r)\) is larger than \(\tilde{w}\), where \(0 < \tilde{w} \leq 1\). The maximum value of \(w(q,r)\) is set as 1.
2. Sampling parameter in low-level layer is 0, i.e., \(p_L = 0\).
3. \(\text{SelectTargetState}\) in Algorithm 1 uniformly samples a random discrete region.

The first condition (c1) assumes that each high-level state has a minimal probability to be selected. The second condition (c2) means that the vertex to extend is selected as the closest vertex to the sampled point; which makes \(\text{ExtendTree}\) (Algorithm 2) similar to traditional sampling based methods [1], [2]. In the proposed method, \(\text{SelectTargetState}\) samples a region of interest with probability \(p_H\) or a feasible random state (with probability \(1 - p_H\)). The last condition (c3) still reflects this property.

In addition, since the use of long extensions does not impair probabilistic completeness and asymptotic optimality [3], we do not mention it separately.

### A. Notation

These are notations that we are using in this material.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\phi)</td>
<td>syntactically co-safe LTL formula</td>
</tr>
<tr>
<td>(A_\phi)</td>
<td>deterministic finite automaton of (\phi), (A_\phi = (Q, \Sigma, \delta, q_{init}, Q_{acc}))</td>
</tr>
<tr>
<td>(D)</td>
<td>graph returned by discrete abstraction, (D = (R_d, E_d))</td>
</tr>
<tr>
<td>(P)</td>
<td>high-level states (A_\phi.Q \times D)</td>
</tr>
</tbody>
</table>

### B. Probabilistic Completeness

Let \(\tilde{R} = r_1, \ldots, r_M\) be a sequence of interest regions \((r_i \in D.R_d)\). A sequence of states \(x = x_1, \ldots, x_M\) , where \(x_i \in r_i\), satisfies the temporal logic formula \(\phi\), which can be represented as \(\text{trace}(x) \models \phi\).

We define a sequence of subsets of \(X\) denoted as \(A_{\tilde{R}} = \{A_0, A_1, A_2, \ldots, A_k\}\), which passes \(\tilde{R}\). \(i = i_1, \ldots, i_M\) is a sequence of indices of \(A_{\tilde{R}}\) with \(M \leq k + 1\). Following properties are hold for \(A_{\tilde{R}}\) and \(i\):

- \(i_j < i_{j+1}, \forall j = 1, \ldots, M - 1\).
- \(i_M = k\).
- \(A_0 = \{x_{init}\}\).
- \(A_{i_j} \subseteq r_j, \forall j = 1, \ldots, M\).

Notice that each adjacent subsets in \(A_{\tilde{R}}\) do not have to be overlapped, and the sequence \(A\) ends with the specific interest region \(A_k \subseteq r_M\). This is natural since atomic propositions of an LTL formula correspond to regions of interest, and trajectory should pass a specific region of interest in order to progress toward an accepting automaton state.

Let \(D(A_i)\) be a set of decomposed regions which cover \(A_i\):

\[
D(A_i) = \{Y_d(x) \mid \forall x \in A_i\}.
\]

For a search tree \(T\), let \(Q_T(A_i)\) be high-level states of vertices in \(A_i\), whose path from the root \(x_{init}\) follows the sequence of subsets \(\{A_0, A_1, \ldots, A_i\}\).

Let \(\tilde{p}\) be defined as

\[
\tilde{p} = \min_{i \in \{0, 1, \ldots, k\}} \min_{r \in D(A_i)} \frac{\mu(A_i \cap Y_d^{-1}(r))}{\mu(X)}.
\]
which corresponds to a lower bound on the probability that a random state will lie in the intersection of particular $A_i$ and one of decomposed region $D(A_i)$.

The following lemma explains the expected number of iterations in the proposed algorithm.

**Lemma 1:** If a sequence $A_{i-1}$ with length $k$ exists, then the expected number of iterations required to pass specific regions of interest $R_i$ (or satisfy $\phi$) is no more than $k/p$, where

\[
p = \frac{\bar{w}}{N_d \cdot N_p^2} \cdot \bar{p}.
\]

**Proof:** If $A_{i-1}$ contains a vertex $v$, then we can compute the probability that $A_i$ will contain a vertex $v'$ connected from $v$, which is denoted as $P(A_i)$. Let $\langle q_s, r_s \rangle$, $\sigma_H$, and $r_t$ are the results of functions $SelectInitialState, DiscretePlanner, SelectTargetRegion$ in Algorithm 1. $P(A_i)$ is now computed as follows:

\[
P(A_i) = P(\langle q_s, r_s \rangle \in Q_T(A_{i-1})) \cdot P(Q_T(A_i) \cap \sigma_H \neq \emptyset).
\]

For each term in $P(A_i)$, we can find the lower bound.

(i) Since the high-level state to be extended is selected by weight $w(q, r)$ and $w(q, r)$ satisfying the condition (c1), the following inequality holds:

\[
P(\langle q_s, r_s \rangle \in Q_T(A_{i-1})) = \frac{\sum_{(q', r') \in Q_T(A_{i-1})} w(q', r')}{\sum_{(q, r) \in P} w(q, r)} \geq \frac{\bar{w}}{N_p}.
\]

(ii) A certain way for high-level plan $\sigma_H$ to include one of $Q_T(A_i)$ is to set it as the target state in the discrete planning procedure. By the condition (c2), we get the following inequality:

\[
P(Q_T(A_i) \cap \sigma_H \neq \emptyset) \geq \frac{1}{N_d |\sigma_H|} \geq \frac{1}{N_p}.
\]

From the above inequalities, the lower bound of $P(A_i)$ can be found as below:

\[
P(A_i) \geq \frac{\bar{w}}{N_d \cdot N_p^2} \cdot \frac{\mu(A_i \cap \bigcap d^{-1}(r_t))}{\mu(\bigcap d^{-1}(r_t))} \geq \frac{\bar{w}}{N_d \cdot N_p^2} \cdot \frac{\mu(A_i)}{\mu(\bigcap)} \geq \frac{\bar{w}}{N_d \cdot N_p^2} \cdot \bar{p} \geq p.
\]

In the worst case, the iterations can be considered as Bernoulli trials with probability of success $p$. A feasible solution is obtained after $k$ successful extension progresses from $A_{i-1}$ to $A_i$. Let $C_1, C_2, \ldots, C_n$ be i.i.d. random variables with the Bernoulli distribution with parameter $p$. The random variable $C = C_1 + C_2 + \ldots + C_n$ is defined as the number of successes after $n$ iterations and $C$ has the binomial distribution

\[
\binom{n}{k} p^k (1-p)^{n-k},
\]

where $k$ is the number of successes. The expected number of iterations for finding path satisfying $\phi$ is $k/p$ ($E[C] = np$).

Since we consider the worst case, this value represents an upper bound.

The following establishes that the probability of failure decreases exponentially with the number of iterations.

**Lemma 2:** If a sequence $A_{i-1}$ with length $k$ exists, the probability that search tree fails to find a path satisfying $\phi$ after $n$ iterations is at most $\exp\left(-\frac{1}{2}(np - 2k)\right)$.

**Proof:** The random variable $C$ in the proof of Lemma 1 has a binomial distribution. A Chernoff-type bound on tail probabilities can be applied to $C$. If $\delta \in (0, 1]$ and $\mu_C = E[C]$, then

\[
P(C \leq (1 - \delta) \cdot \mu_C) < \exp\left(\frac{\mu_C - \delta^2}{2}\right).
\]

By setting $\delta = 1 - k/(np)$ and $\mu_C = np$,

\[
\exp\left(\frac{\mu_C - \delta^2}{2}\right) = \exp\left(\frac{-np}{2} + k - \frac{k^2}{2np}\right) = \exp\left(\frac{-np - 2k}{2} \cdot \exp\left(-\frac{k^2}{2np}\right)\right).
\]

Since $\exp\left(-\frac{k^2}{2np}\right) \leq 1$, we get the following result:

\[
P(C \leq k) < \exp\left(-\frac{1}{2}(np - 2k)\right).
\]

The state at time $t + \Delta t$ is determined as

\[
x(t + \Delta t) = x(t) + \int_t^{t+\Delta t} f(x(t), u(t))dt
\]

for $u(t)$. Let $\bar{u}$ be the correct input for state $x$. Since $f$ is a smooth (continuously differentiable) function and

\[
\dot{x} = f(x, u) = \frac{dx}{dt} \approx \frac{x(t + \Delta t) - x(t)}{\Delta t},
\]

for small amount of time $\Delta t$, the state after input $\bar{u}$ is

\[
x = x(t + \Delta t) = x + \Delta tf(x, \bar{u}) + H_1,
\]

where $H_1$ is high order terms. If a perturbed input $\bar{u} + \delta$ is applied to the system for small $\delta$, then the state $x'$ at time $t + \Delta t$ is as follows with high-order terms $H_2$:

\[
x' = x + \Delta tf(x, \bar{u} + \delta) + H_2.
\]

Since $|H_1 - H_2|$ is negligible for small $\Delta t$, we can find $\delta$ such that $\|x' - \bar{x}\| < \epsilon$. Define a set for $\delta$ as $D(\delta) = \{\delta' \in U \mid \|\delta'\| \leq \|\delta\|\}$. If $U$ is bounded, there exists a positive probability at least $\mu(D(\delta)) / \mu(U) > 0$ such that a randomly selected input can steer state $x$ into $B(\bar{x})$.

We now prove the probabilistic completeness of proposed method.

**Theorem 1:** Algorithm 1 is probabilistically complete.
Proof: If there exists a trajectory satisfying $\phi$, an appropriate $A_R$ with length $k$ can be defined according to the trajectory. Let $x_i$ be a vertex in $A_i$, for $i = 0, \ldots, k - 1$. Assume that the tree $T$ contains $x_i$ as a vertex after some finite number of iterations. There exists a positive probability $p_1$ for extending $x_i$ toward $x_{i+1}$. Also, by positive probability $p_2$ depending on $p_1$, correct input $u_i$ is selected (even for random input when $U$ is bounded). Both $x_i$ and $u_i$ have a probability at least $p_2$ to be selected. Therefore, from Lemma 2, the probability of finding a solution trajectory tends to 1 as the number of iterations increases.

C. Asymptotic Optimality

We state two assumptions from [4], which are modified to our problem. First, we assume local controllability of the system defined as follows.

Assumption 1: There exist constants $\alpha, \epsilon \in \mathbb{R}^+$, $p \in \mathbb{N}$, such that for any $e \in (0, \epsilon)$, and any state $z \in X$, the set $R_e(z)$ contains a ball of radius $\alpha e^p$.

Second, we present the hypothesis about the environment, such as obstacles and regions of interest, to ensure that there exists an optimal trajectory satisfying the given temporal logic specification with enough free space around it.

Assumption 2: There exist an optimal feasible trajectory $\Xi^* : [0, T^*] \to X_{\text{free}}$, constants $\kappa \in \mathbb{R}^+$, $p \in \mathbb{N}$, and a continuous function $q : \mathbb{R}^+ \to X$ with $\lim_{\epsilon \to 0} q(\epsilon) = \Xi^*$ such that for all $e \in (0, \epsilon)$ following two properties hold for the path $\Xi^* = q(\epsilon)$:

- $\Xi^*$ has strong $e$-clearance [2] and satisfies the LTL formula $\phi$.
- For any $t_1 < t_2$, let $z_1 = \Xi^*(t_1)$ and $z_2 = \Xi^*(t_2)$, then $B_{e/\|z_1-z_2\|^p}(z_2) \subset R_e(z_1)$ for some $p \geq 1$.

The function $q$ ensures the existence of a class of paths near the optimal trajectory $\Xi^*$, starting from the initial state and satisfying the given LTL specification. The constants $\alpha$ and $p$ in the second property of Assumption 2 are assumptions on the dynamics, so that such trajectories can be generated from the given dynamic $f$.

We define the last assumption on each decomposed region $D.R_d$.

Assumption 3: For each decomposed region $r \in D.R_d$ and a state $x \in Y^{-1}_d(r)$, there exists a constant $\kappa$ in $(0, 1]$ such that the following inequality holds:

$$\mu(B_{e}(x) \cap Y^{-1}_d(r)) \geq \kappa \cdot \mu(B_{e}(x)), $$

where $e \in \mathbb{R}^+$. Let $\sigma^*_H$ be the corresponding high-level plan of $\Xi^*$. Assuming that the optimal low-cost trajectory between any pair of points is unique, $\sigma^*_H$ is unique since $\Xi^*$ is the minimum cost solution. Consider two adjacent high-level states $\sigma^*_{H,1}, \sigma^*_{H,2} \in \sigma^*_H$ and corresponding two vertices $v_1, v_2$ in the search tree $T$. There exists a trajectory segment $\Xi^*_{1,2} \subset \Xi^*$, connecting $v_1$ and $v_2$ optimally. Let $\Xi^*_{1,2}$ be the trajectory from $T$, which connects $B_{\epsilon}(v_1)$ and $B_{\epsilon}(v_2)$ with $r_\epsilon = \gamma \left(\frac{\log(n)}{n}\right)^{1/d}$, where $n$ is the number of vertices in $T$.

The following lemma shows that the proposed method can search $\Xi^*_{1,2}$ as iteration proceeds.

**Lemma 3:** For sufficiently large $\gamma$, the difference in cost between $\Xi^*_{1,2}$ and $\Xi^*_1$ goes to 0, as $n \to \infty$.

Proof: By conditions (c1) and (c3), there exists a positive probability that high-level states $\sigma^*_{H,1}$ and $\sigma^*_{H,2}$ include vertices of the tree $T$, which also means that the existence of vertices near $v_1$ and $v_2$, respectively. During the proof, vertices in high-level states $\sigma^*_{H,1}$ and $\sigma^*_{H,2}$ are considered, and we assume that the number of vertices and iteration number is equal for simplicity.

We define a function $q_1, q_2(\epsilon)$ for $\Xi^*_{1,2}$ which satisfies properties in Assumption 2 beside satisfying the LTL formula $\lim_{\epsilon \to 0} q_1, q_2(\epsilon) = \Xi^*_1$. For each $n \in \mathbb{N}$ and the selected $\epsilon_n \in (0, \epsilon)$, a sequence of overlapping balls that cover $q_1, q_2(\epsilon_n)$ is constructed as $B_n = \{B_{n,1}, B_{n,2}, \ldots, B_{n,M_n}\}$.

Each ball in $B_n$ has a radius of $r_n = \gamma \left(\frac{\log(n)}{n}\right)^{1/d}$, and centers are separated by the distance $l^{1/p}$, where $l = \beta \epsilon_n$ for constant $\beta$. $\epsilon$ and $p$ are defined in Assumption 2. Since the balls are overlapped each other, the distance between centers is bounded $l^{1/p} \leq 2r_n$.

For two adjacent balls in $B_n$, whose centers are $x_1$ and $x_2$, Assumption 1 guarantees that $R_e(x_1)$ has a positive volume, and there exists a constant $\alpha$ such that $B_{\alpha e/\|x_1-x_2\|^p}(x_2) \subset R_e(x_1)$ for any $e \in (0, \epsilon)$ following two properties hold for the path $\Xi^*_1 = q_1(\epsilon)$:

- $\Xi^*_1$ has strong $e$-clearance [2] and satisfies the LTL formula $\phi$.
- For any $t_1 < t_2$, let $z_1 = \Xi^*_1(t_1)$ and $z_2 = \Xi^*_1(t_2)$, then $B_{\alpha e/\|z_1-z_2\|^p}(z_2) \subset R_e(z_1)$ for some $p \geq 1$.

The function $q_1$ ensures the existence of a class of paths near the optimal trajectory $\Xi^*_1$, starting from the initial state and satisfying the given LTL specification. The constants $\alpha$ and $p$ in the second property of Assumption 2 are assumptions on the dynamics, so that such trajectories can be generated from the given dynamic $f$.

If the proposed method sample vertices inside all balls in $B_n$, it will return a path which has a cost close to that of $q_1(\epsilon_n)$ since the cost function $c$ is continuous and bounded.

Let us assume enough large $n$ so that $\beta \epsilon_n \leq l$ satisfies with some constant $\beta$. An event $F_n$ denotes that no vertices are included in $B_n$ at iteration $n$, and $P(F_n)$ for its probability. As $n \to \infty$, $P(F_n)$ can be upper-bounded by the product of the number of balls that cover the trajectory $q_1, q_2(\epsilon_n)$ and the probability that a ball in $B_n$ does not contain a vertex. If $L_i$ is the length of trajectory of $q_1, q_2(\epsilon_n)$, then $L_i^{1/p}$ is an approximate number of balls in $B_n$. From Lemma 1 and Assumption 3, the minimum probability that a ball $B_{n,i} \in B_n$ contains a vertex for the single iteration is proportional to $\mu(B_{n,1})/\mu(X)$, where $\mu(B_{n,1}) = \pi_d \gamma d \left(\frac{\log(n)}{n}\right)$ with the volume of unit sphere $\pi_d$ in $d$ dimensions. Now $P(F_n)$ satisfies the following inequality:

$$P(F_n) \leq \gamma_1 \cdot L_i^{1/p} \cdot \left(1 - \frac{\gamma^d \log(n)}{\gamma_2 n}\right)^n, \quad (1)$$

where $\gamma_1$ and $\gamma_2$ are constants ($\gamma_1, \gamma_2 \in \mathbb{R}^+$). From $\beta \epsilon_n = \beta' \cdot \gamma \left(\frac{\log(n)}{n}\right)^{1/d} \leq l$, the inequality (1) can be written as

$$P(F_n) \leq \frac{\gamma_1}{\beta l^{1/p} \gamma_1^{1/p}} \cdot \left(1 - \frac{\gamma^d \log(n)}{\gamma_2 n}\right)^n. \quad (2)$$

From $x \leq \exp(x - 1)$,

$$\left(1 - \frac{\gamma^d \log(n)}{\gamma_2 n}\right)^n \leq \left(\exp \left(-\frac{\gamma^d \log(n)}{\gamma_2 n}\right)\right)^n \leq \exp \left(-\frac{\gamma^d \log(n)}{\gamma_2 n}\right) \leq n^{-\gamma^d/\gamma_2}.$$
Apply the above result to the inequality (2), we have
\[ P(F_n) \leq \frac{\gamma_1}{\beta^{1/p} \rho_1^{1/p}} \cdot \frac{1}{\left(\log n\right)^{d/p}} \cdot n^{-\gamma/d} \]
\[ \leq G \cdot n^{-\gamma/d} + d/p, \]
where \( G \) is a constant. Since \( \sum_{n=1}^{\infty} P(F_n) < \infty \) for sufficiently large \( \gamma \), the event \( F_n \) can occur only finitely often due to the Borel-Cantelli lemma [5]. Therefore, as iteration proceeds, the generated trajectory \( \Xi_{1,2} \) becomes close to \( q_{1,2}(\epsilon_n) \), which means approaching to \( \Xi^*_{1,2} \).

From Lemma 3, we state the asymptotic optimality of the proposed algorithm.

**Theorem 2:** Algorithm 1 is asymptotically optimal.

**Proof:** CARL returns a solution if there exists a vertex whose automaton state belongs to an accepting state in \( A_o, Q_{acc} \). Hence, the solution trajectory satisfies the LTL specification. The high-level planner selects the initial high-level states randomly and considers all possible next high-level states with positive probability. Since the number of high-level states is finite, all feasible transitions are selected infinitely often asymptotically. The trajectory segment from the search tree, which connects two vertices of two adjacent high-level states in \( \sigma_H^* \), becomes the optimal low-cost trajectory as iteration proceeds. Therefore, as iteration goes, the cost between the solution trajectory \( \Xi \) and the optimal trajectory \( \Xi^* \) becomes 0, while the corresponding high-level plan is \( \sigma_H^* \).

II. EXPERIMENTAL RESULTS

In this section, we provide additional simulation results which are omitted in the paper due to the page limitation.

A. Convergence

In the paper, we only shows the convergence of the proposed method for an LTL formula for a sequential mission. We consider two additional missions:
\[ \phi_1 = \diamond(a) \land \diamond(b) \land \diamond(c) \land \diamond(d) \]
\[ \phi_2 = \diamond(a \land (w \lor b) \land (w \lor v) \land (u \lor c))), \]
where \( \phi_1 \) is a coverage mission of four discrete regions and \( \phi_2 \) is a strict sequential mission with \( w \) denoting the workspace beside regions of interest. \( \phi_1 \) represents for “cover regions a, b, c and d”, and \( \phi_2 \) for “visit regions with the strict order: a, b and c”.

The results are shown in Figure 1, where brighter regions are higher cost areas. The cost of solution trajectory converges to the optimal cost as the iteration number increases.

B. Strategic Planning for Surveillance

In the paper (section V-B), we compared proposed algorithm with other sampling-based path planning algorithms with LTL constraints [6], [7]. Additional simulation results are stated in below.

1) Computation time: In Figure 2, we show the cost and computation time for generating the first trajectory, i.e., a single run of each algorithm. The algorithm [7] is fast to find a solution and the proposed method requires a longer computation time to find the first solution trajectory due to its long-extension and rewiring steps. However, the cost of the trajectory found by the proposed method is lower than other approaches.

2) Effect of parameters: The proposed method is based on the following parameters: \( \alpha, \beta, p_H, \) and \( p_L \). \( \alpha \) and \( \beta \) are parameters used to determine the weight of a high-level state, where \( \alpha \) controls the effect of the coverage of vertices and \( \beta \) on the frequency of high-level state visitations. \( p_H \) and \( p_L \) are planning parameters for the high-level layer and the low-level layer, respectively. We have tested how each parameter affects the performance of the algorithm. Figure 3 shows simulation results. Default values of parameters are \( \alpha = 1, \beta = 1, p_H = 0.6, \) and \( p_L = 0.2 \). Ten independent runs are performed for each configuration. Notice that the parameter \( \alpha \) has almost no effect on the average performance of the proposed method. In general, we find that there is no single parameter configuration which works the best and the performance of the proposed method is not on average dominated by the parameter configuration.

3) Different cost measure: To demonstrate the effectiveness of the proposed method, we have tested the case when the cost of a trajectory is the length of the trajectory. This cost function is generally used by most existing methods. Simulations are tested with planning parameters \( p_H = 0.8, p_L = 0.2 \), and results are shown in Figure 4. The proposed method and ML-RRT*, a variation of the proposed method without long extension, still outperform [6], [7]. This is due to the rewiring procedure which is required to find a trajectory with the minimum length. However, the performance difference between the proposed method and other algorithms are reduced. In Scenario 3, ML-RRT* shows the best performance since the effect of long extensions is not significant when the cost is the length of a trajectory.

REFERENCES

Fig. 1. (a) A solution found by the proposed method for $\phi_1$. (c) A solution found by the proposed method for $\phi_2$. The initial states are marked by blue squares. (b,d) The cost of a trajectory found by the proposed method (in red) as a function of the iteration number. The optimal cost is shown in blue.

Fig. 2. Simulation results showing the cost and computation time (in seconds) of the first trajectory found by each algorithm. The average value of each algorithm is computed from 15 independent trials and one standard deviation is shown as an error bar.
Fig. 3. Simulation results showing average trajectory costs as a function of running time (in second) with different parameter settings.

Fig. 4. Simulation results showing average trajectory cost for four scenarios. The average cost of each algorithm is computed from 15 independent runs and one standard deviation is shown as an error bar.